

UNIT 9



python™

File, Exception

張傑帆 Chang, Jie-Fan



OUTLINE

- 檔案
- 例外



檔案處理

- 從檔案讀進資料或將結果存入檔案之中
- 從檔案讀進資料
 - `file.read()`
 - `file.readline()`
 - `file.readlines()`
- 將結果存入檔案
 - `file.write()`
 - `file.writelines()`



檔案

- 使用 **open()** 函數來開啟電腦中已經存在的檔案，以便能進行檔案內容的讀取、寫入或修改等動作



- 而使用方式如下所示：

```
file object = open(file_name [, access_mode][, buffering] )
```

檔案

```
file object = open(file_name [, access_mode][, buffering] )
```

- `[]` 裡面的參數都是可省略的
- **file_name** 就是想開啟的檔案，其中也包含了檔案的路徑，如果沒有包含路徑的話，則要開啟的檔案與目前在編輯的程式碼檔都位於同一個資料夾
- **access_mode** 想要存取的模式，種類繁多，下面分別列表說明
- **buffering** 資料讀入的暫存空間
 - 0 代表沒有暫存空間，大於或等於1 則是代表有暫存空間，及一次可讀入多少資料量，例如，3 就是表示一次讀入3 個固定大小的chunk的資料量。至於-1，則表示會用預設(4096-8192 bytes)的暫存區大小來暫存所讀入的資料內容。



檔案

模式	描述
r	以唯讀方式打開文件。文件游標將會放在文件的開頭。預設模式。
r+	用於讀寫。文件游標將會放在文件的開頭。
w	功能只限定於寫入。如果該文件已存在則將其覆蓋。如果該文件不存在，創建新文件。
w+	能夠讀寫。如果該文件已存在則將其覆蓋。如果該文件不存在，創建新文件。
a	功能限定為追加資料。文件游標將會放在文件的結尾，新的內容將會被寫入到已有內容之後。如果該文件不存在，創建新文件進行寫入。
a+	能夠讀寫。文件游標將會放在文件的結尾。文件打開時會是追加模式。如果該文件不存在，創建新文件用於讀寫。



模式	描述
rb	以 唯讀 方式打開文件(二進制 格式)。文件游標將會放在文件的 開頭 。 預設模式 。
rb+	以 二進制 格式打開一個文件用於 讀寫 。文件游標將會放在文件的 開頭 。
wb	(二進位 格式)功能只限定於 寫入 。如果該文件已存在則將其 覆蓋 。如果該文件不存在， 創建 新文件。
wb+	(以 二進位 格式)能夠 讀寫 。如果該文件已存在則將其 覆蓋 。如果該文件不存在， 創建 新文件。
ab	(以 二進位 格式且功能限定為 追加資料 。如果該文件已存在，文件游標將會放在文件的 結尾 。也就是說，新的內容將會被寫入到已有內容 之後 。如果該文件不存在， 創建 新文件進行寫入。
ab+	(以 二進位 格式)功能限定為 追加資料 。如果該文件已存在，文件游標將會放在文件的 結尾 。如果該文件不存在， 創建 新文件並且能夠 讀寫 。



檔案

- 用 **open** 開檔之後要執行的動作至少有以下三種：
- **讀檔**，**寫檔**及**關檔**，以**read()**、**write()**、**close()** 為代表
 - 讀檔有**read()**，**readline()**，**readlines()** 三種
 - 寫檔有**write()**、**writeline()**、**writelines()** 三種
 - 關檔就有**close()** 一種
- **read()** 一次可以**讀取整份文件**，通常會將整份文件的文字放在一個**string** 裡面，對於文件中「行」的概念處理起來相當麻煩，因此在讀檔時並不常使用
- **readlines()** 一次讀取整份文件後將文件依行為單位**存成 list**，接下來就可以使用**for 迴圈**進行處理
- **readline()** **一次只讀取一行**，會比較慢。如果記憶體的空间足夠，建議使用**readlines()**



01	f = open('text.txt', 'r')	# 以讀方式打開檔
02	for line in f.readlines():	# 依次讀取每行
03	line = line.strip()	# 去掉每行頭尾空白
04	print(line)	
05	f.close()	

>>>

1

Stray birds of summer come to my window to sing and fly away.
And yellow leaves of autumn, which have no songs, flutter and fall there
with a sign.

2

A Troupe of little vagrants of the world, leave your footprints in my
words.

...

10

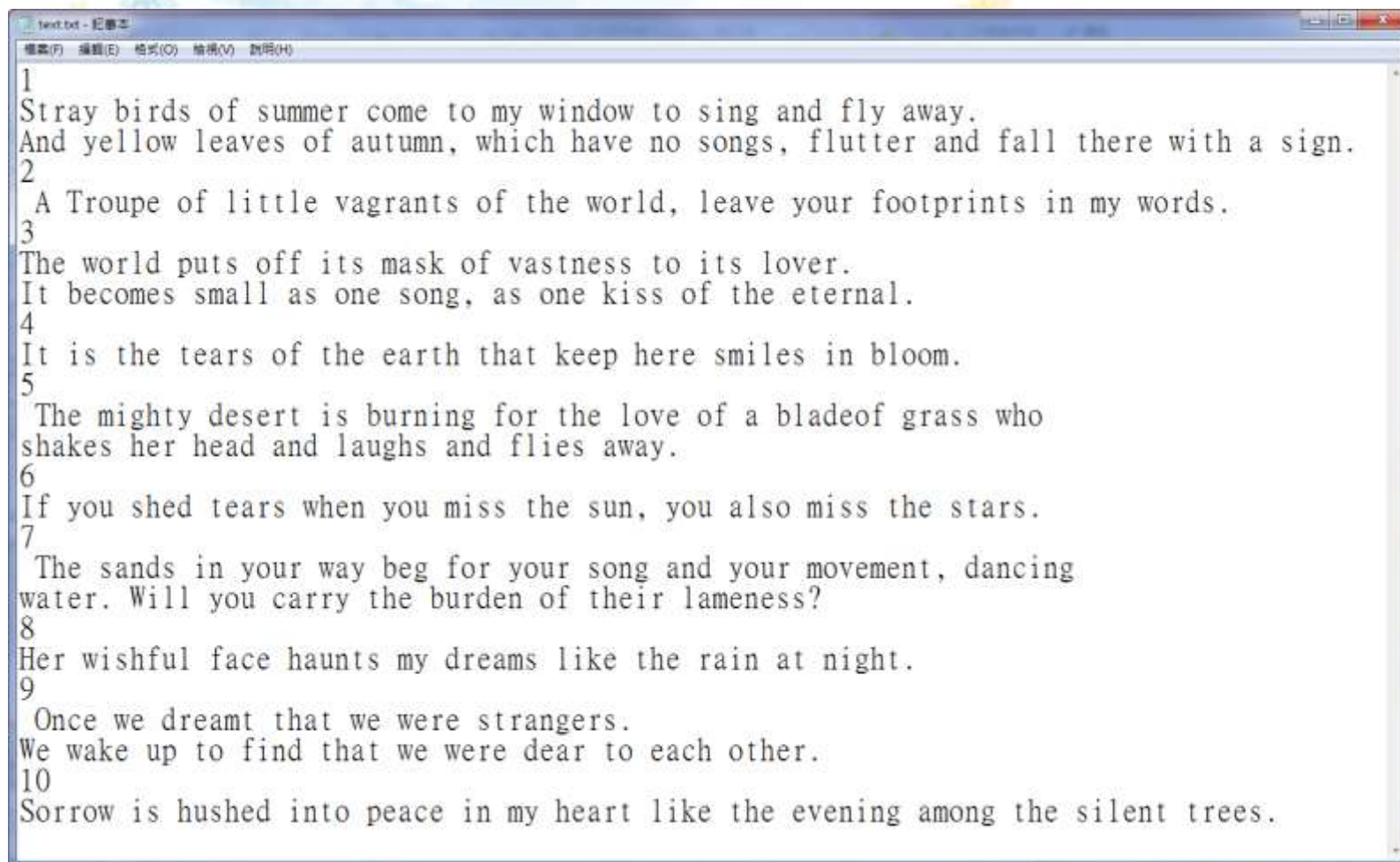
Sorrow is hushed into peace in my heart like the evening among the
silent trees.

>>>



檔案

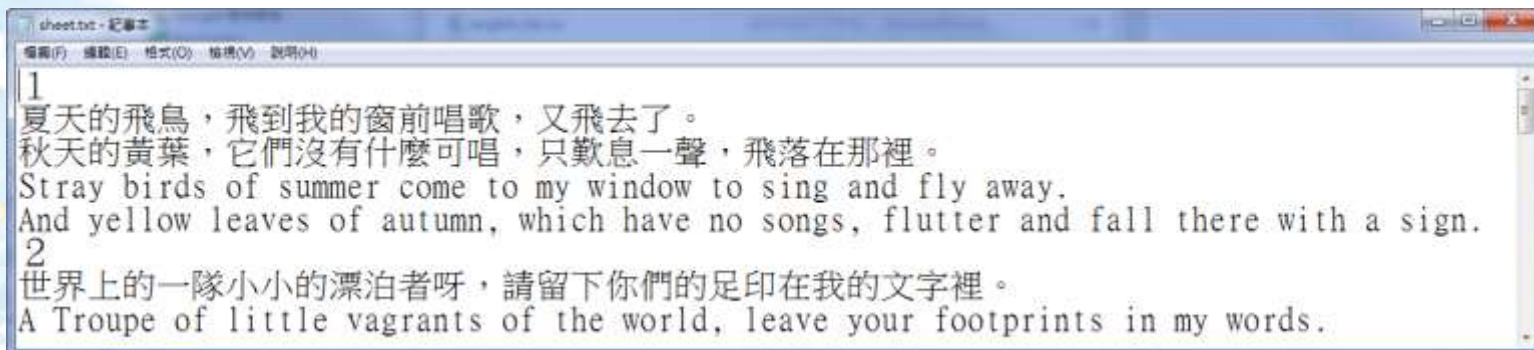
■ 文字檔案內容



1
Stray birds of summer come to my window to sing and fly away.
And yellow leaves of autumn, which have no songs, flutter and fall there with a sign.
2
A Troupe of little vagrants of the world, leave your footprints in my words.
3
The world puts off its mask of vastness to its lover.
It becomes small as one song, as one kiss of the eternal.
4
It is the tears of the earth that keep here smiles in bloom.
5
The mighty desert is burning for the love of a blade of grass who
shakes her head and laughs and flies away.
6
If you shed tears when you miss the sun, you also miss the stars.
7
The sands in your way beg for your song and your movement, dancing
water. Will you carry the burden of their lameness?
8
Her wishful face haunts my dreams like the rain at night.
9
Once we dreamt that we were strangers.
We wake up to find that we were dear to each other.
10
Sorrow is hushed into peace in my heart like the evening among the silent trees.



檔案



```
01 f = open("sheet.txt", "r")           # 以唯獨模式開啟檔案
02 print(f.readline())
03 print(f.readline())                 # 單行讀入
04 print(f.readline())
05 print(f.readline())
06 print(f.readline())
07 f.close()                          # 關閉檔案
```

>>>

1

夏天的飛鳥，飛到我的窗前唱歌，又飛去了。

秋天的黃葉，它們沒有什麼可唱，只歎息一聲，飛落在那裡。

Stray birds of summer come to my window to sing and fly away.

And yellow leaves of autumn, which have no songs, flutter and fall there with a sign.

>>>



小練習

- 讀取指定檔名之文字檔案
- <https://jgirl.ddns.net/problem/0/1206>
- 輸入一個字串表示檔案的名字。
例如 **x.txt**
- 從 **x.txt** 文字檔中讀一字串到**str**
- 再將其所讀到的字串列印出來
- 文字檔中有可能會有空白字元。如下
- **asdfqwer 1234#\$%^&*(**



相對路徑與絕對路徑

- 相對路徑 相對於現在目錄的路徑表示法，因此「**相對路徑**」所指到的檔案或目錄，會**隨著現在目錄的不同而改變**
- 假設當前目錄於E:\test\file
 - ./ 表示**當前路徑**，相當於E:\test\file
 - ../ 表示當前路徑的**上一級路徑**，相當於E:\test
 - ./data 表示當前路徑**下一級路徑**裡要的**data**資料夾裡，相當於E:\test\file\data



相對路徑與絕對路徑

- 絕對路徑

- 指的是一個絕對的位置，並不會隨著現在目錄的改變而改變

- EX:

- E:\test
- E:\test\file
- E:\test\file\data



讀取檔案

- `name = input('請輸入檔名：')`
`file = open(name, 'r', encoding='UTF-8')`
`content = file.read()`
`print(content)`
`file.close()`
- **read()** 方法會一次讀取所有的檔案內容，在不使用檔案時，可以使用 **close()** 將檔案關閉以節省資源。



編碼

- 舉凡所有數位的資訊都是由**0和1**所組合而成的，包括音訊、影像、文字全都是。
- 那為什麼我們在電腦螢幕上看到的不是0跟1呢？
- 那是因為電腦經過「**編碼**」這個動作，把0跟1轉換成人類所看得懂的**文字**及前面所提及的**聲音**與**影像**等等。



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



亞洲地區編碼大亂鬥

Japanese JIS Level1 and 2 (Excerpt)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
88Ax	啞	娃	阿	哀	愛	挨	始	逢	葵	茜	穉	惡	握	渥	旭	葦	B0Ax
88Bx	芦	鱗	梓	压	幹	扱	宛	姐	虻	飴	絢	綾	鮎	或	粟	裕	B0Bx
88Cx	安	庵	按	暗	案	闇	鞍	杏	以	伊	位	依	偉	困	夷	委	B0Cx
88Dx	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃	萎	衣	B0Dx
88Ex	謂	違	遣	医	井	亥	域	育	郁	磯	一	壹	溢	逸	稻	茨	B0Ex
88Fx	芋	鰓	允	印	咽	員	因	姻	引	飲	淫	胤	蔭				B0Fx

Korean KSC5601(Excerpt)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	가	각	간	관	갈	갈	감	감	갑	갑	갓	갓	강	갯	갯
갈	갈	갈	개	객	간	갈	감	갑	갓	갓	갠	갠	각	간	갈
갯	갯	개	간	갠	거	격	건	건	걸	검	검	검	것	것	경
것	걸	걸	걸	게	겐	겔	겜	겜	겟	겟	겜	겨	격	격	견
견	걸	검	검	것	것	경	겔	계	겐	겔	겜	겟	고	곡	곤
곤	골	골	골	골	곰	곰	곳	공	곳	과	과	관	관	관	

Simplified Chinese GB2312(Excerpt)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
B0Ax		啊	阿	埃	挨	哎	唉	哀	皑	癌	蔼	矮	艾	碍	爱	隘	B0Ax
B0Bx	鞍	氨	安	俺	按	暗	岸	胺	案	肮	昂	盎	凹	敖	熬	翱	B0Bx
B0Cx	袄	傲	奥	懊	澳	芭	捌	扒	叭	吧	笆	八	疤	巴	拔	跋	B0Cx
B0Dx	靶	把	耙	坝	霸	罢	爸	白	柏	百	摆	佰	败	拜	裨	斑	B0Dx
B0Ex	班	搬	扳	般	颁	板	版	扮	拌	伴	瓣	半	办	绊	邦	帮	B0Ex
B0Fx	梆	榜	膀	绑	棒	磅	蚌	镑	傍	谤	苞	胞	包	褒	剥		B0Fx

Traditional Chinese Big-5(Excerpt)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	陞	陝	除	陞	陞	隻	飢	馬	骨	高	鬥	鬲	鬼	乾	偕
偽	停	假	偃	佞	做	偉	健	偶	偃	偕	偵	側	偷	偏	倏
儗	倂	兜	冕	凰	剪	副	勒	務	勘	動	匐	匏	匙	匿	區
區	參	曼	商	咄	啦	啄	啞	啡	啃	啊	唱	啖	問	啣	唯
啤	唸	售	啜	唬	啣	啖	啞	啞	圈	國	圍	域	堅	罌	堆
埠	埠	基	堂	堵	執	培	夠	奢	娶	婁	婉	婦	婪	炯	



- <https://www.microsoft.com/hk/hkscs/code/>
- <http://blog.xievc.com/common-code-standard-unicode-utf-iso-8859-1-etc/>



編碼

- 在所有程式語言中，編碼都是相當重要的一環。
- 如果編碼錯誤：
「鎖愼纓緄」潛灝塚丰紉嬪紡瑗悃■甯場娇璽一殯
瑾炆■，錫屾櫟鉏〃彌間嫫斃絳梲暉」
- 這樣的文字即為「亂碼」，就是因為編碼錯誤所造成的。



編碼

- 在電腦裡面，繁體中文編碼1 主要分成以下兩種：**BIG-5** 以及**UTF-8**。
- 在**Python** 的世界裡這兩種都支援，但是這裡比較推薦**UTF-8**。
 - **BIG-5** 是專門拿來作為繁體中文的編碼方式，對於其他語系的系統相容性較低；
 - **UTF-8** 是符合**Unicode** 格式的可變長度字元編碼，可以把它看成是電腦與各種不同語系之間的翻譯橋梁。



編碼

- 如果一支程式可以解讀**UTF-8**，那麼對於各種語系的解碼也就不成問題了。
- 在**Python 3.x** 版以後的版本也將**UTF-8** 設為**預設編碼**，因此即便不手動執行編碼的動作也可以使用中文；
- 但是如果您想要使用**Python 2.x** 的版本或者想要使用其他的編碼方式，那麼就必須要另外進行設定了。



編碼

`#-*-coding:UTF-8-*-`

- 上面看到的這個就是標準的Python 編碼方式，而其詳細說明則分述如下：「#」是開始註解的意思。
- 而註解是給程式撰寫者看的，電腦在執行程式時會自動跳過註解。
- 但是當「#」後面接「coding: 編碼方式」時，整句陳述就變成宣告編碼的意思，此時電腦便不會跳過，而是將其解讀為宣告編碼。



編碼

- 「`coding:UTF-8`」就是跟電腦說我要使用UTF-8的編碼方式。
 - 當然，如果想要換成其他編碼方式，只要把「UTF-8」換成您想要的編碼方式就可以了。
- 最初跟最後的「`-*-`」是跟電腦講說編碼開始與結束的意思。
 - 不過，隨著Python版本的不斷演進，現在編碼連`-*-`都可以省去了，直接`#coding:`就可以了。



編碼

- 從標準的編碼方式可以延伸出比較簡略的方式，以下分別列出並比對說明：

<code>#-*-coding:UTF-8-*-</code>	# 正式寫法，Python 編碼聲明格式
<code>#coding:UTF-8</code>	# 這樣也可以，可捨去-*-
<code>#encoding:UTF-8</code>	# 這樣也可以，coding 亦可使用encoding
<code>#coding= UTF-8</code>	# 這樣也可以，可用=取代：
<code># coding: UTF-8</code>	# 這樣也可以，# 後可容許空格。
	# 注意：2.7 版coding 與: 或= 間不容許空格。
	# 3.3 版之後是允許可以有空格的。



編碼

- 繁體中文自1980年代以來通常使用Big5中文編碼
- 簡體中文，傳統上使用GB2312、GBK或GB18030編碼
- **UTF-8 (8-bit Unicode Transformation Format)**
是一種針對Unicode的可變長度字元編碼，也是一種字首碼。它可以用來表示Unicode標準中的任何字元，且其編碼中的第一個位元組仍與ASCII相容，這使得原來處理ASCII字元的軟體無須或只須做少部份修改，即可繼續使用。因此，它逐漸成為電子郵件、網頁及其他儲存或傳送文字的應用中，優先採用的編碼。



讀取其它編碼的檔案

- 讀取../app/sim.txt檔
- 並將所有內容給列印出來
- 須將encoding設為簡體中文編碼

1	010杜甫：佳人
2	絕代有佳人，幽居在空谷。
3	自云良家子，零落依草木。
4	關中昔喪亂，兄弟遭殺戮。
5	官高何足論，不得收骨肉。
6	世情惡衰歇，萬事隨轉燭。
7	夫婿輕薄兒，新人美如玉。
8	合昏尚知時，鴛鴦不獨宿。
9	但見新人笑，那聞舊人哭！
10	在山泉水清，出山泉水濁。
11	侍婢賣珠回，牽蘿補茅屋。
12	摘花不插發，采柏動盈掬。
13	天寒翠袖薄，日暮倚修竹。

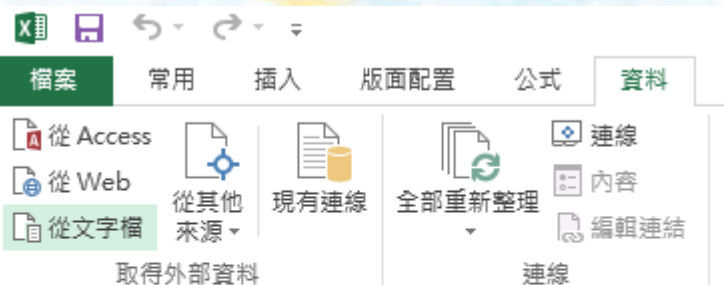


CSV檔

- Comma-Separated Values (逗號分隔值)
- 檔案以純文字形式儲存表格資料（數字和文字）
- 是一種被分隔的資料格式，它有被逗號字元分隔的欄位/列和以換行結束的記錄/行。
- 由任意數目的記錄組成，每條記錄由欄位組成，欄位間的分隔符號，最常見的是逗號或制表符號
- CSV檔案格式的通用標準並不存在，但是7-bit ASCII是最基本的通用編碼。



EXCEL開CSV檔亂碼解決



匯入字串精靈 - 步驟 3 之 1

資料剖析精靈判定資料類型為 分隔符號。

若一切設定無誤，請選取 [下一步]，或選取適當的資料類別。

原始資料類型

請選擇最適合剖析您的資料的檔案類型：

- ☒ 分隔符號 (D) — 用分欄字元，如逗號或 TAB 鍵，區分每一個欄位。
- ☐ 固定寬度 (W) — 每個欄位固定，欄位間以空格區分。

起始列號 (R): 1 檔案原始格式 (O): 65001 : Unicode (UTF-8)

預覽檔案 C:\Users\user\Downloads\政府資料開放平臺資料集清單.csv

1 "資料集名稱", "檔案格式", "下載連結", "資料集類型", "資料集描述", "主要欄位說明", "資料集提供機
2 "蒙藏委員會補助團體私人情形季報表", "csv", "http://www.mtac.gov.tw/pages/283/MTAC005.csv", "
3 "蒙藏用語對照表", "csv", "http://www.mtac.gov.tw/pages/283/MTAC004.zip", "rawData", "蒙藏用語
4 "蒙藏用語對照表", "zip", "http://www.mtac.gov.tw/pages/283/MTAC004.zip", "other", "蒙藏用語對

取消

< 上一步 (B)

下一步 (N) >

完成 (F)

匯入字串精靈 - 步驟 3 之 2

您可在畫面中選擇輸入資料中所包含的分隔符號，您可在預覽視窗內看到分欄的結果。

分隔符號

☒ Tab 鍵 (T)

☐ 分號 (M)

☐ 逗點 (C)

☐ 空格 (S)

☒ 其他 (O): ,

☐ 連續分隔符號視為單一處理 (R)

文字辨識符號 (Q): "

預覽分欄結果 (P)

資料集名稱	檔案格式	下載連結	資料
蒙藏委員會補助團體私人情形季報表	csv	http://www.mtac.gov.tw/pages/283/MTAC005.csv	rawD
蒙藏用語對照表	csv	http://www.mtac.gov.tw/pages/283/MTAC004.zip	rawD
蒙藏用語對照表	zip	http://www.mtac.gov.tw/pages/283/MTAC004.zip	othe

取消

< 上一步 (B)

下一步 (N) >

完成 (F)

名稱

修改日期

政府資料開放平臺資料集清單

2015/6/12 上午 0...



NTU CSIE

檔案名稱 (N): 政府資料開放平臺資料集清單

文字檔案

工具 (L)

匯入 (M)

取消

小練習

- 試著使用 `readlines` 讀取一 `*.csv` 檔後將其印出

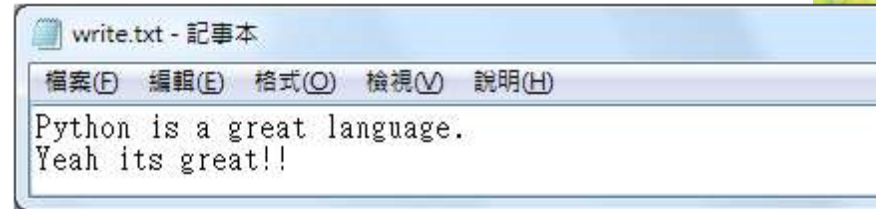


- `write()` 方法可將任何字串寫入目前已經被打開的檔案
- `write()` 方法不會自動在字串的結尾中添加分行符號（`'\n'`）
- 因此如果要換行則必須自行加上分行的符號

```
01 # 打開一個檔
02 fo = open("write.txt", "w+")
03 fo.write("Python is a great language.\nYeah its great!!\n")
05 fo.close()      # 關閉打開的文件
07 # 執行完上面的程式碼，會在這個Python 檔案的路徑中新建一個
   write.txt
```

```
09 # 下面要再開啟這個檔，並且輸出這個檔的內容
```

```
10 fo = open("write.txt", "r")
11 for line in fo.readlines():
12     line = line.strip()
13     print(line)
14 fo.close()
```



```
>>>
```

```
Python is a great language.
```

```
Yeah its great!!
```

```
>>>
```



檔案

- **writeline()**
 - 輸入完之後會換行
- **writelines()**
 - 則是將**list** 當作輸入參數，以**list** 的資料型態將資料寫入檔案
 - 當**list** 中每個元素被寫入後，並不會自動換行



小練習 I

- 1. 讀取*.CSV檔
- 2. 直接將讀到的*.CSV檔
寫入另一*.CSV文字檔中

■



小練習 I-PLUS

- 將**csv**的每個欄位填入**list**中，並切割成為二維**list**，以供查詢
- 例如：請列出第一間門市的門市名稱：
- `print(txt[1][3])` 會列出「海景門市」
- 使用語法提示：
 - `str.strip()`
 - `str.split()`
- 1. 請試印出所有的門市名稱
- 2. 將上述**list**寫到另一新的**csv**檔中



進階練習 II

- 讀取 stores_old.csv 檔案後，將 sid,name,tel,wifi 篩選出來並寫入檔案至 stores_new.csv 中 (要分隔「,」)

A	B	C	D	E	F	G	H
sid	longitude	latitude	name	address	tel	wifi	note
1	121.740089	25.132571	海晏門市	200基隆市仁愛區港西路6之2號	02-2408-4223	Y	
2	121.51797	25.047029	新泰祥門市	100台北市中正區忠孝西路一段49號1樓	02-2371-7092		
3	121.509098	25.041583	豐平門市	100台北市中正區寶通路32號	02-2331-0388	Y	
4	121.531439	25.017797	羅新福門市	106台北市大安區羅斯福路三段301號	02-2364-6845		
5	121.514167	25.046001	德華門市	100台北市中正區康寧路10號	02-2394-3776	Y	
6	121.57694	25.062673	聯華門市	114台北市內湖區舊港路一段188號一樓	02-2752-4408	Y	
7	121.563288	25.041236	聯合門市	110台北市信義區忠孝東路四段561號1樓	02-2749-5657	Y	
8	121.548296	25.051753	理亞門市	105台北市松山區南京東路三段337號1F	02-2712-0625	Y	
9	121.566807	25.082169	聯山門市	114台北市內湖區內湖路一段289號	02-2799-5897	N	
10	121.540795	25.06212	龍華門市	104台北市中山區民權東路三段22號	02-2517-0259	Y	
11	121.548336	25.041369	龍門門市	106台北市大安區忠孝東路四段134號	02-2740-4782	Y	
12	121.543721	25.03332	南門門市	106台北市大安區復興南路一段323號1樓	02-2325-0473	Y	
13	121.544467	25.034355	興光門市	105台北市松山區復興北路147號1、2F	02-2712-8189	Y	
14	121.557866	25.048104	德源門市	105台北市松山區八德路三段251號1-2F	2598-4752	Y	
15	121.507392	25.043915	漢中門市	100台北市萬華區漢中街51號1-3樓	02-2370-5293	Y	
16	121.542392	25.083801	富源門市	104台北市中山區大直街70號1樓	02-2532-5233	Y	
17	121.553019	25.041263	南豐門市	106台北市大安區忠孝東路四段218號一樓	2773-7207	Y	
18	121.557369	24.999536	萬祥門市	116台北市內湖區瑞興路三段111號	02-2933-6950	Y	
19	121.569521	25.078441	瑞源門市	114台北市內湖區瑞光路476號	02-2788-3266	Y	
20	121.572749	25.077284	瑞光門市	114台北市內湖區瑞光路356號	02-2659-3912	Y	
21	121.567366	25.039564	新光門市	110台北市信義區松山路11號1樓	02-2789-4099	N	
22	121.520707	25.118416	新光門市	111台北市士林區天母路33號1樓	02-2876-4551	N	
23	121.547395	25.132201	聯南門市	112台北市北投區湖山路一段14號2、3樓	02-2881-0399	Y	
24	121.526934	25.043442	華南門市	106台北市大安區光復南路100號4樓	02-2771-9692	Y	
25	121.575919	25.075292	瑞源門市	114台北市內湖區瑞光路116號	02-2753-5105	Y	

stores_old.csv

A	B	C	D	E	F
sid	name	address	tel	wifi	note
1	海晏門市	200基隆市仁愛區港西路6之2號		Y	
2	新泰祥門市	100台北市中正區忠孝西路一段49號1樓			
3	豐平門市	100台北市中正區寶通路32號		Y	
4	羅新福門市	106台北市大安區羅斯福路三段301號			
5	德華門市	100台北市中正區康寧路10號		Y	
6	聯華門市	114台北市內湖區舊港路一段188號一樓		Y	
7	聯合門市	110台北市信義區忠孝東路四段561號1樓		Y	
8	理亞門市	105台北市松山區南京東路三段337號1F		Y	
9	聯山門市	114台北市內湖區內湖路一段289號		N	
10	龍華門市	104台北市中山區民權東路三段22號		Y	
11	龍門門市	106台北市大安區忠孝東路四段134號		Y	
12	南門門市	106台北市大安區復興南路一段323號1樓		Y	
13	興光門市	105台北市松山區復興北路147號1、2F		Y	
14	德源門市	105台北市松山區八德路三段251號1-2F		Y	
15	漢中門市	100台北市萬華區漢中街51號1-3樓		Y	
16	富源門市	104台北市中山區大直街70號1樓		Y	
17	萬祥門市	116台北市內湖區瑞興路三段111號		Y	
18	瑞源門市	114台北市內湖區瑞光路476號		Y	
19	瑞光門市	114台北市內湖區瑞光路356號		Y	
20	新光門市	110台北市信義區松山路11號1樓		N	
21	新光門市	111台北市士林區天母路33號1樓		N	
22	聯南門市	112台北市北投區湖山路一段14號2、3樓		Y	
23	華南門市	106台北市大安區光復南路100號4樓		Y	
24	瑞源門市	114台北市內湖區瑞光路116號		Y	

stores_new.csv



進階練習 III

- 讀取 `stores_old.csv` 檔案後
- 小練習 1:
請列出 **公館門市** 的所有資訊
- 小練習 2:
令使用者輸入一 **門市名稱**，並列出該門市的所有資訊



(回家) 小練習 IV

- 讀取 `stores_old.csv` 檔案後
- 列出所有有wifi的門市的所有資訊

```
1,121.740089,25.132571,海景門市,200基隆市仁愛區港西街6之2號,02-2428-4223,Y,  
3,121.509098,25.041583,寶平門市,100台北市中正區寶慶路32號,02-2331-0388,Y,  
4,121.531439,25.017797,羅斯福門市,106台北市大安區羅斯福路三段301號,02-2364-6845,Y,  
...  
208,121.156972,24.043806,清境門市,南投縣仁愛鄉定遠新村26-1號1樓,049-280-3798,Y,  
209,120.910651,23.86684,日月潭門市,555南投縣魚池鄉中山路101號,049-285-6849,Y,  
210,119.569217,23.566203,馬公門市,880澎湖縣馬公市三民路2號,06-927-3459,Y,
```



延伸閱讀

- **csv 套件**
 - [CSV File Reading and Writing](#)
- **Excel 套件**
 - [Working with Excel Files in Python](#)
 - [Working with Excel Spreadsheets](#)
- **Pandas 套件讀 csv**
 - https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html
- **JSON 格式**
 - <https://docs.python.org/3/library/json.html>
- **yaml 格式**
 - <http://angeloce.iteye.com/blog/385976>



PICKLE 變數存檔

saveVar.py

```
1 import pickle
2
3 dict = {'da':111, 2:[23,1,4], '23':{1:2,'d':'sad'}}
4
5 # pickle a variable to a file
6 file = open('pickle_example.pickle', 'wb')
7 pickle.dump(dict, file)
8 file.close()
9
```

loadVar.py

```
1 # reload a file to a variable
2 with open('pickle_example.pickle', 'rb') as
3 file:
4     dict1 =pickle.load(file)
5 print(dict1)
```



例外

- 程式碼在執行時，也可能會發生錯誤
- 這代表著在設計這支程式的時候，有些地方不甚周詳
- 或是在執行這支程式時，發生了**意想不到的錯誤**
- 如果**不希望程式就此終止**
- 就需要將可預期的錯誤另外撰寫程式來預防
- 使用**try...except**。
- 至少要存在一個**except** 敘述
- **else:** 跟**finally:**則是可有可無

try:

欲執行的程式碼T

except 例外情形1 [參數]:

欲執行的程式碼1

except 例外情形2 [參數]:

欲執行的程式碼2

...

except 例外情形N [參數]:

欲執行的程式碼N

[else:

欲執行的程式碼E]

[finally:

欲執行的程式碼F]



為什麼需要例外處理

■ 除以0

```
a = int(input())  
b = int(input())
```

```
print(a/b)  
print("hello python")
```

```
10  
0  
Traceback (most recent call last):  
  File "J:/NTU CSIE Train/錄影/ex/PYT  
<module>  
    print(a/b)  
ZeroDivisionError: division by zero
```



加入例外處理後

```
a = int(input())
```

```
b = int(input())
```

```
try:
```

```
    print(a/b)
```

```
except:
```

```
    print("例外發生，不可除以0")
```

```
print("hello python")
```

```
10
```

```
0
```

```
例外發生，不可除以0
```

```
hello python
```

```
>>>
```



例外

- 如果當try 裡面的程式碼執行時發生異常，Python 就跳回到try 的起始位置，並執行第一個符合該異常的except 敘述。等異常處理完畢後，程式流程就算執行完整個try 區塊中的程式碼，除非在處理異常時又引發新的異常
- 如果在try 裡面的程式碼裡發生了異常，卻沒有匹配的except 敘述，異常將被轉送到上一層的try 區塊處理，或者到程式的最上層。若是遇到後者情況，將結束程式，並印出錯誤資訊。
- 如果在try 的程式碼執行並無引發異常，如果有else: 的區塊內容的話，Python 便會去執行else: 區塊內部的程式碼
- 最後執行finally: 區塊內部的程式碼，如此便算順利執行完整個try...except 指令區(不管有沒有異常，都會執行)



例外處理

- **try-except** 也可以和 **else** 連用，**else** 後的程式區塊放的是沒有發生例外，程式所執行的工作，例如

```
a = int(input())
```

```
b = int(input())
```

```
try:
```

```
    print(a/b)
```

```
except:
```

```
    print("例外發生，不可除以0")
```

```
else:
```

```
    print("沒有例外發生")
```

```
finally:
```

```
    print("hello python")
```

```
print("錯誤處理結束")
```

```
>>>
```

```
10
```

```
0
```

```
例外發生，不可除以0
```

```
hello python
```

```
錯誤處理結束
```

```
>>>
```

```
5
```

```
3
```

```
1.6666666666666667
```

```
沒有例外發生
```

```
hello python
```

```
錯誤處理結束
```

```
>>>
```



例外

- 以下舉例說明：

```
01 i = 10
02 for j in range(3, -4, -1):
03     print('%d/%d=%0.3f'%(i, j, i / j))
```

```
>>>
```

```
10/3=3.333
```

```
10/2=5.000
```

```
10/1=10.000
```

Traceback (most recent call last):

File "C:\Users\John\Desktop\Google 雲端硬碟\Python Wizard\examples\Ch9\ex09_05.py", line 3, in <module>

```
    print('%d/%d=%0.3f'%(i, j, i / j))
```

ZeroDivisionError: division by zero

```
>>>
```



例外

■ 示範程式碼

```
01 i = 10
02 for j in range(3, -4, -1):
03     try:
04         print('%d/%d=%0.3f'%(i, j, i / j))
05     except ZeroDivisionError:
06         print('除數為0，無法進行除法')
```

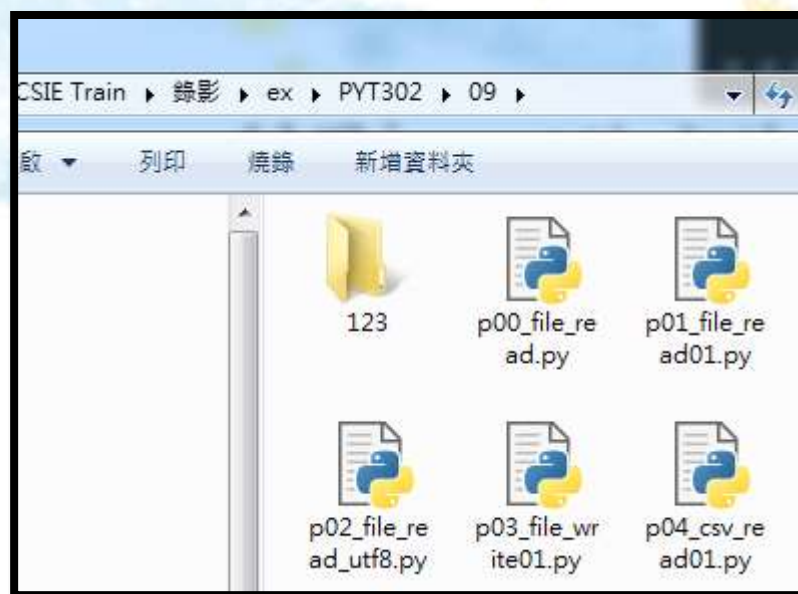
```
>>>
10/3=3.333
10/2=5.000
10/1=10.000
除數為0，無法進行除法
10/-1=-10.000
10/-2=-5.000
10/-3=-3.333
>>>
```



例外

- 如果沒有加上錯誤的類型，則所產生任何錯誤都會被歸類到某個特定的**except** 語句來處理。
- 好處是沒有錯誤會被遺漏，程式一定可以正常執行完畢
- 壞處是如此一來，便無法得知發生什麼錯誤之外也無法針對特定錯誤進行不同的處理，也就沒有辦法輕易地修改程式碼來避免錯誤的發生
- 建議最好不要把**except** 後面的錯誤類型留白，或只留在最後一個，並加上錯誤資訊(見最後頁)
- Python 有內建相當多的錯誤類型，每一種對應的情況都不同，像上面範例所示範的**ZeroDivisionError** 便是專門對應 0 在除數的除法錯誤
- 另外，還有對應**type** 錯誤的**TypeError**、對應導入模組錯誤的**ImportError** 等等，大部分都可以從錯誤名稱看出錯誤的原因





```
>>> import os
>>> os.mkdir('123')
>>> os.mkdir('123')
Traceback (most recent call last):
  File "<pyshell#25>", line 1, in <module>
    os.mkdir('123')
FileExistsError: [WinError 183] 當檔案已存在時，無法建立該檔案。: '123'
>>> os.mkdir(123)
Traceback (most recent call last):
  File "<pyshell#26>", line 1, in <module>
    os.mkdir(123)
TypeError: mkdir: path should be string, bytes or os.PathLike, not int
>>>
```


例外處理

- 例外的名稱也可以寫在 **except** 之後，如

```
a = 10
```

```
b = 0
```

```
try:
```

```
    print(a/b)
```

```
except NameError:
```

```
    print("例外發生，NameError")
```

```
except ZeroDivisionError:
```

```
    print("例外發生，不可除以0")
```

```
print("hello python")
```



例外處理

- **try-except** 也可以和 **else** 連用，**else** 後的程式區塊放的是沒有發生例外，程式所執行的工作，例如

```
a = 10
```

```
b = 0
```

```
try:
```

```
    print(a/b)
```

```
except NameError:
```

```
    print("例外發生，NameError")
```

```
except ZeroDivisionError:
```

```
    print("例外發生，不可除以0")
```

```
else:
```

```
    print("沒有例外發生")
```

```
print("hello python")
```



■ 確保程式能完整執行完畢

```
01 try:
02     print(" 進入try 區塊")
03     for i in range(5, -5, -1):
04         print("10 / " + str(i) + " = ", end = "")
05         print(str(10/i))
06 except:
07     print(" 發生錯誤，進入except 區塊")
08     print(" (except) i = " + str(i))
09 else:
10     print(" 沒有錯誤，進入else 區塊")
11     print(" (else) i = " + str(i))
12 finally:
13     print(" 進入finally 區塊")
14     print(" (finally) i = " + str(i))
```

>>>

進入try 區塊

10 / 5 = 2.0

10 / 4 = 2.5

10 / 3 = 3.3333333333333335

10 / 2 = 5.0

10 / 1 = 10.0

10 / 0 = 發生錯誤，進入except 區塊

(except) i = 0

進入finally 區塊

(finally) i = 0

>>>



課堂練習

```
try:
```

```
a = eval(input("輸入數字a:"))  
b = int(input("輸入數字b:"))  
c = a/b
```

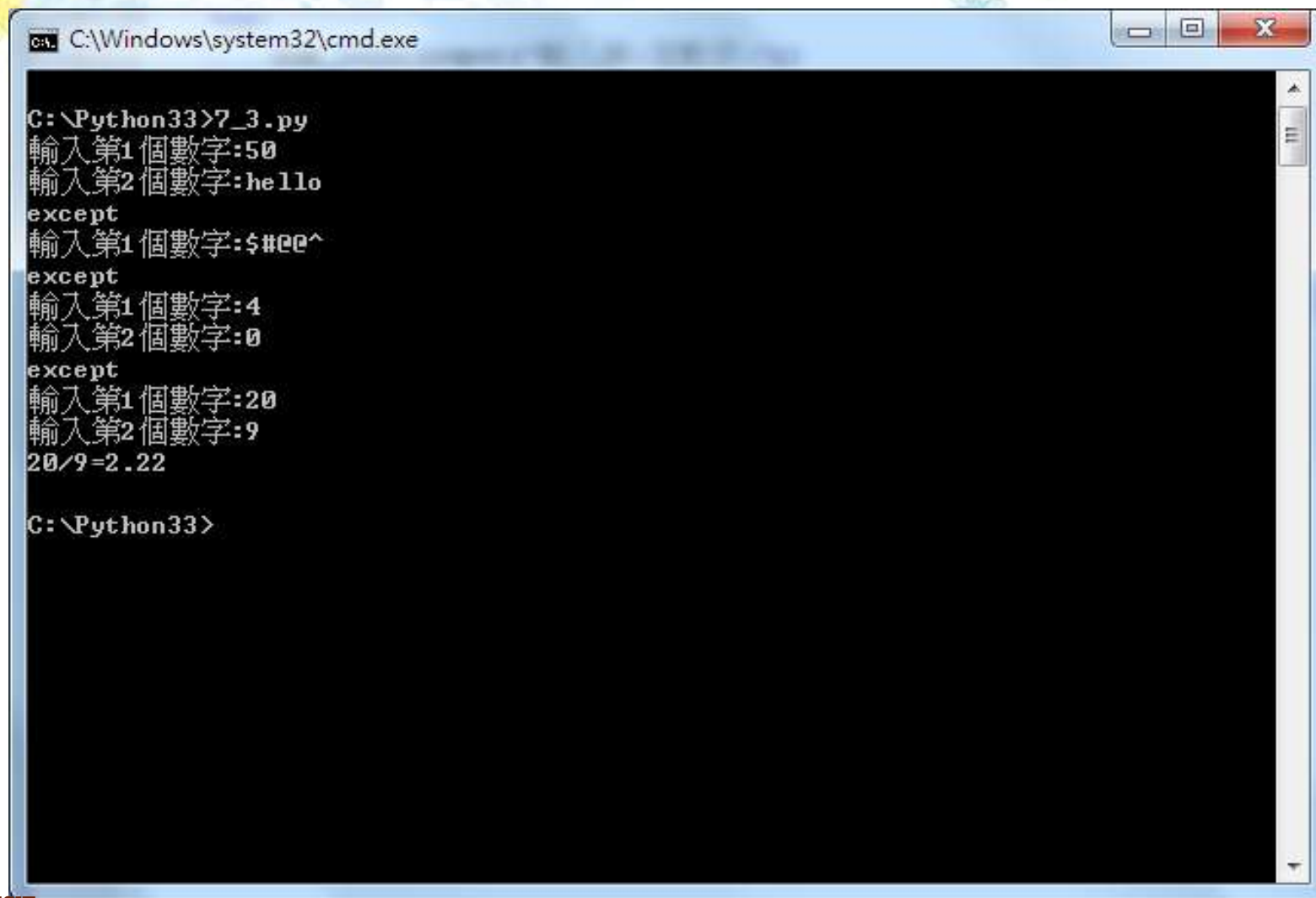
```
.....
```

- 利用 **input** 的方法，輸入兩個數字，並將兩個數字相除的結果印出。
- 加入例外處理機制，若發生任何例外則讓使用者重新輸入直到正確將相除的結果印出才終止程式

```
>>>  
輸入數字a:hello  
NameError  
輸入數字a:10  
輸入數字b:a  
ValueError  
輸入數字a:10  
輸入數字b:0  
ZeroDivisionError  
輸入數字a:5  
輸入數字b:3  
5/3 = 1.67  
>>>
```



執行範例



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The command prompt shows the execution of a Python script named "7_3.py" in the directory "C:\Python33". The script prompts for two inputs: "輸入第1個數字:" and "輸入第2個數字:". It then performs a division operation and displays the result. The execution flow is as follows:

```
C:\Python33>7_3.py
輸入第1個數字:50
輸入第2個數字:hello
except
輸入第1個數字:$#ee^
except
輸入第1個數字:4
輸入第2個數字:0
except
輸入第1個數字:20
輸入第2個數字:9
20/9=2.22

C:\Python33>
```



RESTART: J:/NTU CSIE Train/錄影/ex/PYT302/09/p06_try02.py
輸入數字a:abc
NameError
輸入數字a:10a
Traceback (most recent call last):
 File "J:/NTU CSIE Train/錄影/ex/PYT302/09/p06_try02.py", line 3, in <module>
 a = eval(input("輸入數字a:"))
 File "<string>", line 1
 10a
 ^
SyntaxError: unexpected EOF while parsing
>>>



例外

- 以下列出Python 常見的錯誤類型：

異常名稱	描述
BaseException	所有異常的基礎類型
SystemExit	編譯器請求退出
KeyboardInterrupt	用戶中斷執行(通常是輸入^C)
Exception	常規錯誤的基礎類型
StopIteration	迭代器沒有更多的值
GeneratorExit	generator發生異常來通知退出
SystemExit	Python 編譯器請求退出
StandardError	所有的內建標準異常的基礎類型
ArithmeticError	所有數值計算錯誤的基礎類型
FloatingPointError	浮點計算錯誤
OverflowError	數值運算超出最大限制



例外

異常名稱	描述
OverflowError	數值運算超出最大限制
ZeroDivisionError	除(或取餘數)零(所有資料類型)
AssertionError	斷言語句失敗
AttributeError	物件沒有這個屬性
EOFError	沒有內建輸入,到達EOF(End Of File) 標記
EnvironmentError	作業系統錯誤的基礎類型
IOError	輸入/輸出操作失敗
OSError	作業系統錯誤
WindowsError	系統調用失敗
ImportError	導入模組/對象失敗
KeyboardInterrupt	用戶中斷執行(通常是輸入 ^C)



例外

異常名稱	描述
LookupError	無效數據查詢的基礎類型
IndexError	序列中沒有此索引(index)
KeyError	集合中沒有這個鍵
MemoryError	內存溢出錯誤(對於Python 編譯器不是致命的)
NameError	未宣告/初始化對象(沒有屬性)
UnboundLocalError	訪問未初始化的本地變數
ReferenceError	Weak reference試圖訪問已經垃圾回收了的對象
RuntimeError	一般的執行階段錯誤
NotImplementedError	尚未實作的方法
SyntaxError	Python 語法錯誤
IndentationError	縮排錯誤



例外

異常名稱	描述
TabError	Tab 和空格混用
SystemError	一般的編譯器系統錯誤
TypeError	對類型無效的操作
ValueError	傳入無效的參數
UnicodeError	Unicode 相關的錯誤
UnicodeDecodeError	Unicode 解碼時的錯誤
UnicodeEncodeError	Unicode 編碼時錯誤
UnicodeTranslateError	Unicode 轉換時錯誤
Warning	警告的基礎類型
DeprecationWarning	關於被棄用的特徵的警告
FutureWarning	關於構造將來語義會有改變的警告



例外

異常名稱	描述
OverflowWarning	溢位警告
PendingDeprecationWarning	關於特性將會被廢棄的警告
RuntimeWarning	可疑的runtime behavior警告
SyntaxWarning	可疑的語法的警告
UserWarning	用戶代碼生成的警告



例外處理 **RAISE**陳述

- 測試例外可用 **raise**

```
try:
```

```
    raise ZeroDivisionError
```

```
except NameError:
```

```
    print("發生例外,NameError")
```

```
except ZeroDivisionError:
```

```
    print("發生例外,ZeroDivisionError")
```

```
print("after exception....")
```



延申閱讀

- 官方文件

- <https://docs.python.org/3/library/exceptions.html>
- <https://docs.python.org/3/tutorial/errors.html>

- **Except** 進階資訊 (出錯的行號... 等)

- <https://stackoverflow.com/questions/1278705/python-when-i-catch-an-exception-how-do-i-get-the-type-file-and-line-number>
- <http://www.runoob.com/python/python-exceptions.html>
- <https://segmentfault.com/a/1190000007736783>



EXCEPT 進階資訊

- 顯示出錯的行號與錯誤說明及檔案

```
import os, sys
```

```
try:
```

```
.....
```

```
except Exception as e:
```

```
    exc_type, exc_obj, exc_tb = sys.exc_info()
```

```
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
```

```
    print(exc_type, exc_obj, fname, exc_tb.tb_lineno)
```

