

111-2 商管程式設計

TA Lab 5



AGENDA

- 考試複習
- 考試提醒
- 考古題練習



Course Feedback

01

考試複習



Course Feedback

What have we learned?

01

變數、型別、input()、print()、print() 中 sep 和 end 的用法

02

條件: if、else、elif

03

迴圈: for、while

04

清單: 一維與二維清單、list operations、split()



Course Feedback

input()

- 讓使用者輸入值
- 括號間可填入給使用者的提示訊息
- 取得的輸入型態必為**字串**
可搭配使用**型別轉換**

print()

- 輸出指定的值, 任何型態皆可
- 一次輸出多值, 可利用**,**分開, 輸出各值之間有一個**空格**
- **sep**: 變更分隔符號
 - 用法 **sep='<你要的分隔符號>'**
- **end**: 變更結尾符號
 - 用法 **end='<你要的結尾符號>'**
- 使用escape characters, 表特殊符號
 - e.g., **'\n'**: 印出換行
 - e.g., **'\t'**: 印出tab



Course Feedback

Type conversion

- `input()`
 - 型別預設為 `string`
 - 如果想得到其他 data type
整數？小數？ → **型別轉換**
- 型別轉換
 - `int()`: 將其他型態的參數轉成整數型態
 - `str()`: 將其他型態的參數轉成字串型態
 - `float()`: 將其他型態的參數轉成浮點數型態
 - **不是**所有型態轉換都可行！
e.g., "Hello world" 轉換為 `int` 或 `float`

```
num = input()
print(type(a))
b = int(input())
print(type(b))
num = float(input())
print(type(c))
```

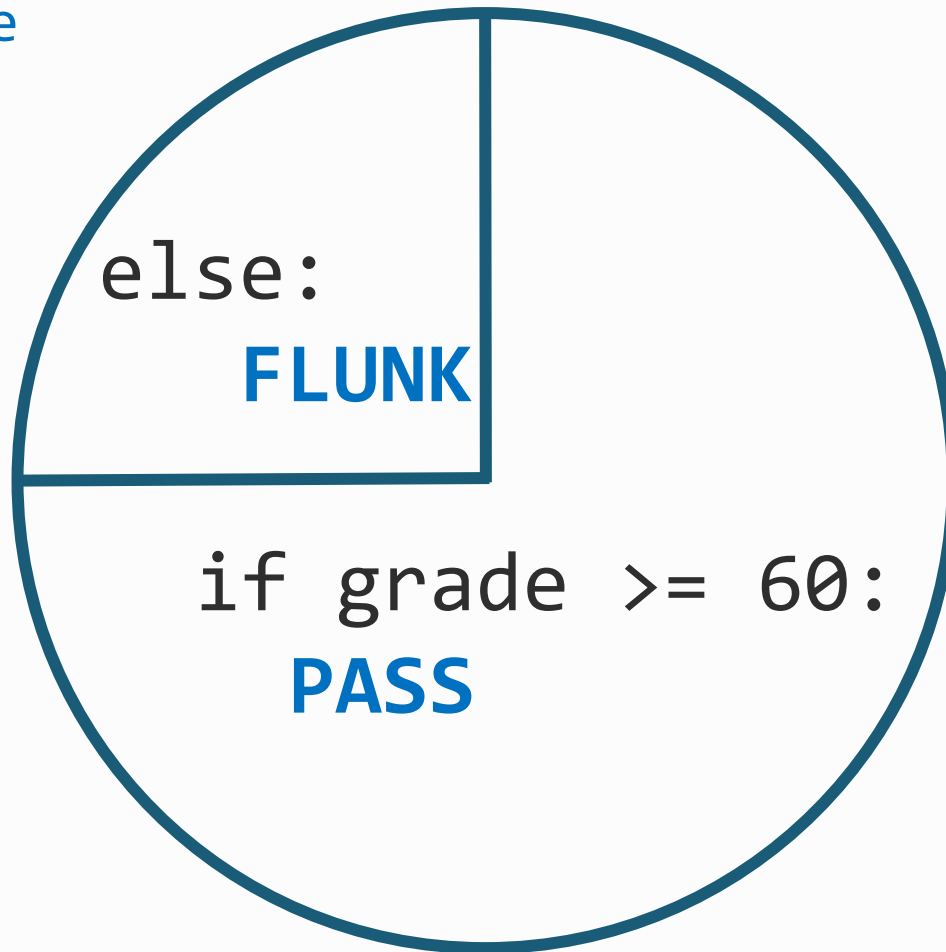
```
a = "52"
b = int(a)
print(a)
print(b)
print(type(a))
print(type(b))
```



Course Feedback

Conditionals

- if-else



```
grade = int(input())  
if grade >= 60:  
    print("PASS")  
else:  
    print("FLUNK")
```

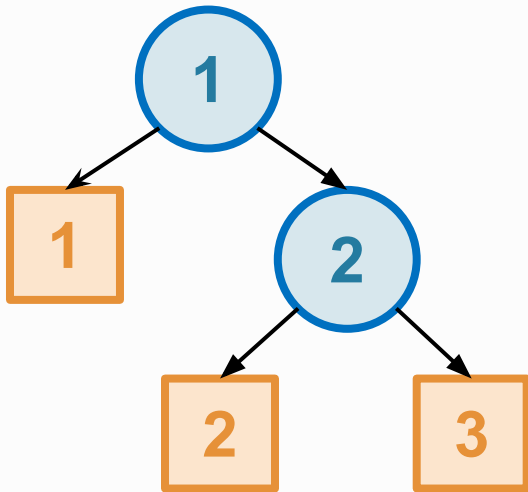


Course Feedback

Nested conditionals

- 巢狀 `if` 判斷式
- 樹狀圖：每組 `if-else` 為一個節點

- `elif`



```
if condition1:
    statement1
else:
    if condition2:
        statement2
    else:
        statement3
```

```
if condition1:
    statement1
elif condition2:
    statement2
else:
    statement3
```




Course Feedback

for loop

- 從**序列**中獲取元素，將元素指派給前面的**迭代變數**
- 指派序列的方法之一：`range()`
 - **`range(start, stop, step)`**

```
for i in range(8):  
    print(i)  
for i in range(1, 8):  
    print(i)  
for i in range(1, 8, 2):  
    print(i)  
for i in range(20, 2, -3):  
    print(i)
```

0 1 2 3 4 5 6 7

1 2 3 4 5 6 7

1 3 5 7

20 17 14 11 8 5

Result



Course Feedback

while loop

- 在**符合某種條件**下，循環執行statement

```
a = 0
while a <= 3:
    a += 1
    print(a)
```

1
2
3
4

Result

- 注意避免**無窮迴圈**

```
a = 0
while a >= 0:
    a += 1
    print(a)
```

Nested loop

- 內圈執行完畢，外圈才會進行下一項

```
for i in range(2):
    for j in range(3):
        print('i:', i, 'j:', j)
```

```
i: 0 j: 0
i: 0 j: 1
i: 0 j: 2
i: 1 j: 0
i: 1 j: 1
i: 1 j: 2
```

Result

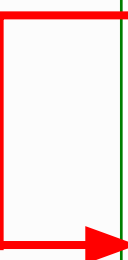


Course Feedback

break

- 強制結束迴圈

```
for i in range(10):  
    if i == 2:  
        break  
    else:  
        print(i)  
print('out of loop')
```




```
0  
1  
out of loop
```

Result

continue

- 強制跳到下一個迴圈

```
i = 0  
while i < 4:  
    i += 1  
    if i == 2:  
        continue  
    else:  
        print(i)  
print('out of loop')
```



```
1  
3  
4  
out of loop
```

Result



Course Feedback

String Concatenation

- 串聯字串: +

```
q = "Quiz:"  
score = 100  
print(q + str(score))  
print(q, score)
```

'Quiz:100'

'Quiz: 100'

Result



String Split

- `str.split('分割符號')`
 - 預設用**空格**切割字串
 - 可以更改成**其他符號**做切割
 - 分割後的型別為 `list`

```
a = "apple banana orange"  
a = a.split()  
print(a)
```

```
['apple', 'banana', 'orange']
```

Result

```
a = "apple,banana,orange"  
a = a.split(',')  
print(a)
```

```
['apple', 'banana', 'orange']
```

Result

```
print(type(a))
```

```
<class 'list'>
```

Result



List Initialization

- 初始化 empty list

```
empty_list_1 = list()
empty_list_2 = []

multiple_list = [0] * 3
print(multiple_list)
# [0, 0, 0]
```

List Operations

- index: 第一個為 0, 最後一個為 -1

方法	功能
<code>.append(x)</code>	將 x 加到 list 最後面
<code>.sort()</code>	將 list 由小到大排列
<code>.reverse()</code>	將 item 從最後一個排到第一個
<code>.index(x)</code>	將 x 第一次出現的 index 回傳
<code>.insert(i,x)</code>	將 x 加到位置 i
<code>.count(x)</code>	算 x 出現幾次
<code>.remove(x)</code>	將第一次出現的 x 刪掉
<code>.pop(i)</code>	將第 i 個 item 刪掉



Course Feedback

List Iteration

- list 為序列, 從中獲取迭代元素

```
answer = [11, 22, 33, 44, 55]
for ans in answer:
    print(ans, end=',')
```

11,22,33,44,55,

Result

- 以 list index 獲取迭代元素

```
for i in range(len(answer)):
    if i < len(answer) - 1: # i < 4
        print(answer[i], end=',')
    else: # i == 4
        print(answer[i])
```

11,22,33,44,55

Result



Course Feedback

程式碼換行

- PEP8 規定，一行程式碼最大長度應限制在 **79** 字元內
 - Limit all lines to a maximum of 79 characters
- 換行方法：使用反斜線 '****'

<pre>a = 1 b = 1 c = a + \ b print(c)</pre>	<div>2</div> <div>Result</div>
---	--------------------------------

02

考試提醒



Course Feedback

期中考注意事項

- 注意考試規定與考場，考場之後會公告，考試位置當天會在教室內公布
- 題目以 NTU COOL 所公布的**pdf 檔**為準
- **Open book, 不能使用通訊軟體**
- 考試當下看不到得幾分，只能看到 Sample 測資有沒有過
- 分數以最後一次上傳為主
- 時間以 PDOGS 系統時間為準
- 避免在最後關頭才上傳答案，以免因為網路不穩影響你的繳交時間
- **只看 PDOGS 有沒有過：不看課內課外、不看 coding style**
- 確定電腦電量足夠
- 大家加油！100100100

03

考古題練習



Course Feedback

110-2 Midterm 1 Problem 1

- <https://pdogs.ntu.im/all-class/2/34/challenge/391/1169>

有一個國家裡有三個城市，分別是首都與另外兩個城市。首都、城市 1、城市 2 分別位在單位為公里的直角座標系 (x_0, y_0) 、 (x_1, y_1) 和 (x_2, y_2) 的位置上，因此首都與城市 1 的直線距離 (Euclidean distance) 是 $\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$ ，與城市 2 的則是 $\sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2}$ 。請找出城市 1 和城市 2 哪個離首都的直線距離比較短，並印出該城市的編號，如果一樣短則印出 0。請注意若比較直線距離，其實不需要開根號，直接比較兩個平方和即可。



Course Feedback

110-2 Midterm 1 Problem 2

- <https://pdogs.ntu.im/all-class/2/34/challenge/391/1170>

有一個國家裡有 $n + 1$ 個城市，分別是首都與另外 n 個城市。首都、城市 1、城市 2 直到城市 n ，分別位在單位為公里的直角座標系 (x_0, y_0) 、 (x_1, y_1) 、 (x_2, y_2) 直到 (x_n, y_n) 的位置上，因此首都與城市 i 的直線距離是 $\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$ 。請找出城市 1 到 n 之中哪個離首都的直線距離比較短，並印出該城市的編號，如果有複數個城市離首都的直線距離同為最短的，則將這些城市的編號由小到大都印出來。請注意若要比較直線距離，其實不需要開根號，直接比較兩個平方和即可。



Course Feedback

110-2 Midterm 1 Problem 3

- <https://pdogs.ntu.im/all-class/2/34/challenge/391/1171>

有一間大型電子商務購物平台商會在使用者瀏覽其頁面時對其投放自家商品的廣告，由於頁面上有三個版位，因此它希望能從自家販售的 n 項商品之中找到前三項最適合被投放的商品，使得該使用者有最大的機率會點擊投放的廣告。該大型電子商務購物平台商所販售的 n 項商品總共分屬 m 個商品類別，商品 i 吸引消費者點擊的機率已經被估算為 p_i ，其所屬之商品類別則為 c_i 。使用者正觀看之產品頁面的產品所屬類別則為 c_0 。請幫平台在所有商品類別為 c_0 的商品中挑出被點擊機率前三高的商品，並由高至低依序印出這三個被點擊機率。



110-2 Midterm 1 Problem 4

- <https://pdogs.ntu.im/all-class/2/34/challenge/391/1172>

有一台機器將進行生產製造，該機器一共會需要完成 n 項工作，工作 i 的作業時間為 p_i 分鐘，工作不可切割（進行中便不可中斷直到完成為止），截止時間為機器開始運作後的第 d_i 分鐘。給定一個排程（工作順序）後，工作 i 的延遲時間（tardiness）為

$$\max\{\text{完成時間} - d_i, 0\},$$

亦即倘若某工作的完成時間超出其截止時間，超出的程度即為延遲時間，但若早於截止時間則延遲時間為 0。排程的目標是最小化有延遲（延遲時間大於零）的工作數。

在本題我們要請你實做一個（簡化過的）叫「最早截止時間優先」（Earliest Due Date, EDD）的演算法（雖然這不一定會產出最佳排程）。首先，將所有工作先依照截止時間由小到大排序，接著嘗試將工作按照這個順序進行。但在嘗試進行一個工作時，要先檢查一下，如果執行該工作會產生延遲，則直接略過該工作不排，繼續嘗試下一個工作直到所有工作都被嘗試過為止。將所有不會造成延遲的工作都排定、執行完後，再將剩下曾被略過的工作依原順序（截止時間由早至晚）做執行。給定 n 個工作，請使用此演算法求得工作的執行順序。



110-1 Midterm 1 Problem 3 (二維清單)

- <https://pdogs.ntu.im/all-class/2/30/challenge/350/964>

你賣 m 種商品，收到 n 張訂單，訂單 i 訂購了商品 j 一共 x_{ij} 個，而商品 j 的售價是 p_j 元，因此訂單 i 的原始總價是

$$P_i = \sum_{j=1}^m p_j x_{ij}$$

元。如果訂單 i 的原始總價不低於 Q 元，你會幫訂單 i 折扣 D 元，但以使最終總價不低於 Q 元為限。換言之，給定 P_i 、 Q 和 D ，訂單 i 的最終總價為

$$\max\{Q, P_i - D\}$$

元。舉例來說，假設你收到了很多張訂單，其中你考慮訂單 i ，如果 $m = 3$ 、 $p_1 = 100$ 、 $p_2 = 200$ 、 $p_3 = 80$ 、 $x_{i,1} = 3$ 、 $x_{i,2} = 2$ 、 $x_{i,3} = 0$ 、 $Q = 500$ 、 $D = 50$ ，則訂單 i 的原始總價為 $100 \times 3 + 200 \times 2 + 80 \times 0 = 700$ 元，不低於 Q 元，且折扣 D 元後的折扣後價格為 $700 - 50 = 650$ 元，不低於 Q 元，所以訂單 i 的最終總價為 650 元。如果其他數字都一樣但 $Q = 680$ ，則最終總價將為 680 元。請幫所有訂單都計算最終總價，並且求出所有訂單的最終總價的總和。

TA Lab 5

課後回饋表單

03.20-03.24

<https://reurl.cc/OVrkQR>

