Python Development More © Enos Chou

Cases

1. 資料結構設計與運用--學生資料儲存、維護，與計算
2. 基礎判斷式--猜拳
3. 基礎巢狀迴圈--完美列印九九乘法表
4. 旗標應用--Missing Number
5. 巢狀迴圈搭配雙層 List 應用--Valid Sudoku
6. 資料結構設計、判斷式、迴圈綜合應用--大老二一決勝負
7. 判斷式、迴圈、演算法綜合應用--成績指標
8. 效率衡量--泡泡排序
9. 檔案處理與分析--Log 分析
10. 檔案處理與分析--CSV 分析
11. 目錄與檔案處理--計算樹木
12. 容錯設計--猜數字
13. 類別設計--大老二莊家
14. 演算法演練--Move Zeros
15. 演算法挑戰--Best Time to Buy and Sell Stock

1. 資料結構設計與運用

學生資料儲存、維護，與計算

Tech

1. Composite data types
2. Operators
3. Loop

Tasks

1. 儲存下列 10 位學生資料，並印出 Bill 的身高體重

   ```
   Bill's height: 168cm, weight: 70kg
   ```
2. 計算每位學生的 BMI 並儲存至上述資料結構
3. 計算並輸出所有學生的 平均 BMI、BMI 變異數、BMI 標準差

   ```
   mean bmi is 23.238 kg/m²
   var bmi is 4.158 (kg/m²)²
   std bmi is 2.039 kg/m²
   ```

Data

```
David    180cm 85kg
John     172cm 72kg
Mary     158cm 51kg
Lora     160cm 49kg
Bill     168cm 70kg
Elsa     154cm 55kg
Golden   176cm 69kg
Jason    182cm 77kg
Larry    160cm 64kg
Michelle 155cm 57kg
```

Hints

1. dict-oriented design
2. list-oriented design

2. 基礎判斷式

猜拳

Tech

1. if-elif-else
2. random
3. input

Tasks

1. 實作猜拳遊戲，讓用戶跟電腦猜拳

3. 基礎巢狀迴圈

完美列印九九乘法表

Tech

1. Nested loop
2. print arguments

Tasks

1. 整齊印出如下的九九乘法表 (注意空格)

Output

```
1 x 1 = 1 2 x 1 =  2 3 x 1 =  3 4 x 1 =  4 5 x 1 =  5 6 x 1 =  6 7 x 1 =
 7 8 x 1 =  8 9 x 1 =  9
1 x 2 = 2 2 x 2 =  4 3 x 2 =  6 4 x 2 =  8 5 x 2 = 10 6 x 2 = 12 7 x 2 =
14 8 x 2 = 16 9 x 2 = 18
1 x 3 = 3 2 x 3 =  6 3 x 3 =  9 4 x 3 = 12 5 x 3 = 15 6 x 3 = 18 7 x 3 =
21 8 x 3 = 24 9 x 3 = 27
1 x 4 = 4 2 x 4 =  8 3 x 4 = 12 4 x 4 = 16 5 x 4 = 20 6 x 4 = 24 7 x 4 =
28 8 x 4 = 32 9 x 4 = 36
1 x 5 = 5 2 x 5 = 10 3 x 5 = 15 4 x 5 = 20 5 x 5 = 25 6 x 5 = 30 7 x 5 =
35 8 x 5 = 40 9 x 5 = 45
1 x 6 = 6 2 x 6 = 12 3 x 6 = 18 4 x 6 = 24 5 x 6 = 30 6 x 6 = 36 7 x 6 =
42 8 x 6 = 48 9 x 6 = 54
1 x 7 = 7 2 x 7 = 14 3 x 7 = 21 4 x 7 = 28 5 x 7 = 35 6 x 7 = 42 7 x 7 =
49 8 x 7 = 56 9 x 7 = 63
1 x 8 = 8 2 x 8 = 16 3 x 8 = 24 4 x 8 = 32 5 x 8 = 40 6 x 8 = 48 7 x 8 =
56 8 x 8 = 64 9 x 8 = 72
1 x 9 = 9 2 x 9 = 18 3 x 9 = 27 4 x 9 = 36 5 x 9 = 45 6 x 9 = 54 7 x 9 =
63 8 x 9 = 72 9 x 9 = 81
```

4. 旗標應用

Missing Number

Tech

1. Loop
2. Dynamic allocation

Tasks

1. 實作 LeetCode > Top Interview Questions > Easy Collection > Others > Missing Number

5. 巢狀迴圈搭配雙層 list 應用

Valid Sudoku

Tech

1. Nested loop

2. Operators

Tasks

1. 實作 LeetCode > Top Interview Questions > Easy Collection > Array > Valid Sudoku

Hints

```
1. //
2. %
```

6. 資料結構設計、判斷式、迴圈綜合應用

大老二一決勝負

Tech

1. if-elif-else
2. Data types
3. Loop
4. random
5. Weights design

Tasks

1. 隨機各發一張撲克牌給 player1 與 player2，撲克牌結構範例為 ('SPADE', '3') 代表 黑桃3
2. 比較 player1 與 player2 勝負
    a. 先比數字，數字大小依序為 '2' > 'A' > 'K' > 'Q' > 'J' > '10' > '9' > '8' > '7' > '6' > '5' > '4' > '3'，較大者獲勝
    b. 若雙方數字相同則比較花色，花色大小依序為 'SPADE' > 'HEART' > 'DIAMOND' > 'CLUB'，較大者獲勝
3. 輸出 player1 與 player2 勝負

7. 判斷式、迴圈、演算法綜合應用

成績指標

Tech

1. if-elif-else
2. Loop
3. Algorithm
4. time

Tasks

1. 完成 APCS 實作題_題目範例：成績指標
2. 找出最有效率的演算法

Hints

1. 執行時間可由執行前與執行後的 time.time() 差值測得
2. 效率比較時須注意公平性

8. 效率衡量

泡泡排序

Tech

1. Loop
2. time

Tasks

1. 實作泡泡排序演算法，與 Python 內建排序工具比較執行效率

9. 檔案處理與分析

Log 分析

Tech

1. open
2. if-elif-else
3. Loop
4. string handling
5. dict

Tasks

1. 讀取 log 檔 goodeploy_log.csv 內容，計算各 function 被叫用的次數。輸出的樣式需參閱 Output，項目順序可不同，但格式需完全相同，請注意對齊

Notes

1. [CoursesFirestore:get_course] 樣式即為 function name
2. Function name 需由程式判斷後讀取，不可將如 CoursesFirestore:get_course 等 function name 字樣寫死在程式中

Output

```
CoursesFirestore:get_course       231 times
LineUsersFirestore:get_user       108 times
CoursesFirestore:get_courses       16 times
LineUsersFirestore:increase        93 times
LineUsersFirestore:update_user     13 times
LineUsersFirestore:meet_criterion   3 times
CoursesFirestore:__init__           2 times
```

10. 檔案處理與分析

CSV 分析

Tech

1. open
2. if-elif-else
3. Loop
4. pandas

Tasks

1. 讀取 orders.csv，找出所有數量大於 10,000 的訂單代號，以 list 形式儲存於 orders
2. 以 pandas 完成上述任務

11. 目錄與檔案處理

計算樹木

Tech

1. Directory
2. File
3. Function
4. Loop
5. glob
6. set
7. sorted
8. Output handling

Tasks

1. 於 trees 目錄中，區分訓練集 (train)、測試集 (test) 計算各樹種 .jpg 檔數量，並將結果輸出至螢幕或 output.csv
2. 在 test 資料夾新增空的資料夾 AA，同時在 train 資料夾新增空的資料夾 ZZ，結果是否依然正確？
3. 建立 function 能夠彈性選擇將結果輸出至螢幕或檔案

Output

```
,AS,BJ,CC,DR,FM,KE,LF,MA,MI,MP,PC,RR,TC,TM
test,35,33,44,31,50,50,33,28,25,28,30,25,32,45
train,96,164,201,105,295,203,150,56,100,68,105,68,62,170
```

Notes

1. 'train', 'test', 'AS', 'BJ'... 等不得寫死，須以程式取得
2. AS: 黑板樹, BJ: 茄冬, CC: 樟樹, DR: 鳳凰木, FM: 榕樹, KE: 台灣欒樹, LF: 楓香, MA: 苦楝, MI: 白千層, MP: 水黃皮, PC: 阿勃勒, RR: 大王椰子, TC: 大葉欖仁, TM: 小葉欖仁

12. 容錯設計

猜數字

Tech

1. Function
2. Loop
3. input
4. str
5. Falut tolerance design

Tasks

1. 製作猜數字遊戲，由電腦隨機出 4 個不同數字讓用戶猜，每次用戶猜測的 4 個數字會由電腦輸出 ?A?B 的結果；每個遊戲在用戶完全猜對 (4A0B) 或猜測 10 次之後結束
2. 程式需自訂 function，並做容錯處理

Output

```
round  1 請猜4個數字 1234
round  1: 0A 1B
round  2 請猜4個數字 245 0
輸入錯誤
round  3 請猜4個數字 99aa 輸入錯誤
round  4 請猜4個數字 2 3 45
```

```
round  4: 0A 2B
round  5 請猜4個數字 2322
輸入錯誤
round  6 請猜4個數字 7013
round  6: 1A 0B
round  7 請猜4個數字 AAAd
輸入錯誤
round  8 請猜4個數字 7592
round  8: 2A 2B
round  9 請猜4個數字 7529
round  9: 4A
```

13. 類別設計

大老二莊家

Tech

1. Class
2. Loop
3. input
4. str

Tasks

1. 設計並建立大老二莊家類別，能夠洗牌，發牌，並能夠依大老二規則比較卡牌大小

Usage

In [ ]:
```
dealer = Dealer()
a, b = dealer.deal(2)
print(f'{a} vs {b}')
print(dealer.win(a, b), 'win')
```

```
('CLUB', '7') vs ('HEART', '9')
('HEART', '9') win
```

Tasks

2. 分別建立莊家與大老二卡牌兩個類別，莊家能夠洗牌，發牌，並以大老二卡牌類別的比較運算子依大老二規則比較大小

Usage

In [ ]:
```
dealer = Dealer()
a, b = dealer.deal(2)
```

```
print(f'{a} vs {b}')
print(f'{a} <  {b}: {a < b}')
print(f'{a} == {b}: {a == b}')
print(f'{a} >= {b}: {a >= b}')
```

```
<DIAMOND, Q> vs <HEART, J>
<DIAMOND, Q> <  <HEART, J>: False
<DIAMOND, Q> == <HEART, J>: False
<DIAMOND, Q> >= <HEART, J>: True
```

Hints

1. `__lt__()`
2. `__le__()`
3. `__gt__()`
4. `__ge__()`
5. `__eq__()`
6. `__len__()`
7. `__str__()`

14. 演算法演練

Move Zeros

Tech

1. Loop
2. list
3. Algorithm

Tasks

1. 實作 LeetCode > Top Interview Questions > Easy Collection > Array > Move Zeros

15. 演算法挑戰

Best Time to Buy and Sell Stock

Tech

1. Loop
2. list
3. Algorithm

Tasks

1. 實作 LeetCode > Top Interview Questions > Easy Collection > Dynamic Programming > Best Time to Buy and Sell Stock

In [ ]: