

[HW] Ch3-2_Lists_v02

1. จงสร้างฟังก์ชันหาค่าเฉลี่ย `average` โดยมีพารามิเตอร์เป็นลิสต์ของตัวเลขใดๆ (ตัวเลขอาจเป็นชนิด `int` และ/หรือ `float`) และคืนค่ากลับเป็นค่าเฉลี่ยของลิสต์นั้นๆ (โดยที่ไม่ใช่ฟังก์ชันหาค่าเฉลี่ย `mean()` ที่มีอยู่แล้วในไพธอน)

เขียนคำตอบลงในช่องว่างที่เตรียมไว้ (ระหว่าง `#BEGIN SOLUTION` และ `### END SOLUTION`) แล้วรันโค้ด ตรวจสอบดูว่าได้ผลลัพธ์ถูกต้องหรือไม่

ผลลัพธ์ที่คาดหวัง

```
average([1,2,3,4,5,6,7,8,9]) = 5.0
average([1.1,2.2,3.3,4.4,5.5]) = 3.3
average([9.93,5.3432,9.43423,1.2344,0.3523]) = 5.258826
```

In [4]: `# เขียนโค้ดด้านล่าง แล้วกด Shift+Enter`

```
def average(list):
    ### BEGIN SOLUTION
    pass
    return sum(list)/len(list)
    ### END SOLUTION

print(average([1,2,3,4,5,6,7,8,9])) # Expected result: 5.0
print(average([1.1,2.2,3.3,4.4,5.5])) # Expected result: 3.3
print(average([9.93,5.3432,9.43423,1.2344,0.3523])) # Expected result: 5.258826
```

```
5.0
3.3
5.258826
```

หลังจากเขียนโค้ดเสร็จ ให้รันเซลล์ต่อไปนี ถ้าสามารถรันได้ (ไม่เกิดเออเรอร์) และแสดงผลลัพธ์เป็น `True` 2 ครั้ง ถือว่า "ผ่าน"

In [6]: `# For Test-run`
`# รันโค้ดต่อไปนี้ ถ้าได้ผลลัพธ์ 'True' 2 ครั้ง ถือว่า "ผ่าน"`
`import numpy as np`
`from datetime import datetime`
`print("@", datetime.now().strftime("%d/%m/%Y %H:%M:%S"))`

`lst1 = np.random.randint(10, size=20)`
`lst2 = 10*np.random.random_sample(20)`

`print("\nTest#1 (int)")`
`print(f"lst1 = {lst1}")`
`print(f"average(lst1) should be {lst1.mean()}. \n{average(lst1) = } \n{average(1`
`print("\nTest#2 (float)")`
`print(f"lst2 = {lst2}")`
`print(f"average(lst2) should be {lst2.mean()}. \n{average(lst2) = } \n{'True' if`

@ 25/02/2024 22:03:55

```
Test#1 (int)
lst1 = [2 1 1 6 9 1 6 0 8 0 4 5 2 0 6 5 5 8 1 4]
average(lst1) should be 3.7.
average(lst1) = 3.7
True
```

```
Test#2 (float)
lst2 = [7.8765625 0.54896528 0.68658883 2.5947038 7.51321832 7.61157852
6.25724134 8.07328386 4.54723525 0.06530907 1.00917019 7.01526373
0.62862597 5.52660741 4.89507295 0.87039749 4.56063251 4.53378401
7.92106692 4.47073741]
average(lst2) should be 4.360302268488958.
average(lst2) = 4.360302268488958
True
```

2. จงเขียนโค้ด โดยกำหนดตัวแปรสตริงชื่อ `newton_quote` เก็บคำพูดของเซอร์ไอแซกนิวตัน

```
newton_quote = 'To any action there is always an opposite
and equal reaction; in other words, the actions of two
bodies upon each other are always equal and always
opposite in direction.'
```

จากนั้น ตัดข้อความดังกล่าวให้เป็นคำๆ เก็บเป็นตัวแปรลิสต์ `word_list` (ไม่รวมเครื่องหมายวรรคตอน (เช่น space, comma (,), semicolon (;)) แล้วแสดงผลลัพธ์ของลิสต์ดังกล่าวโดยใช้คำสั่ง `print(word_list, '\n')`

โดยเขียนคำตอบลงในช่องว่างที่เตรียมไว้ (ระหว่าง `#BEGIN SOLUTION` และ `### END SOLUTION`)

ถ้าสามารถรันได้คำสั่งหลัง `### END SOLUTION` ได้โดยไม่เกิดเออเรอร์ และแสดงผลลัพธ์เป็น `True` ถือว่า "ผ่าน"

```
In [11]: # เขียนโค้ดระหว่าง #BEGIN SOLUTION และ ### END SOLUTION แล้วกด Shift+Enter
#BEGIN SOLUTION
newton_quote = 'To any action there is always an opposite and equal reaction; in
newton_quote = newton_quote.replace(';','')
newton_quote = newton_quote.replace(',','')
word_list = newton_quote.split(' ')
print(word_list, '\n')

### END SOLUTION

print(word_list == ['To', 'any', 'action', 'there', 'is', 'always', 'an', 'oppo
equal', 'reaction', 'in', 'other', 'words', 'the', 'actions', 'c
'bodies', 'upon', 'each', 'other', 'are', 'always', 'equal', 'and
'opposite', 'in', 'direction.'])
```

```
['To', 'any', 'action', 'there', 'is', 'always', 'an', 'opposite', 'and', 'equa
l', 'reaction', 'in', 'other', 'words', 'the', 'actions', 'of', 'two', 'bodies',
'upon', 'each', 'other', 'are', 'always', 'equal', 'and', 'always', 'opposite',
'in', 'direction.']
```

True
