

16 Aug 2019

ANL 251 Python Programming

A toy application

(Supplementary readings)

Kevin Kam Fung Yuen, PhD

Senior Lecturer

School of Business

Singapore University of Social Sciences

kfyuen@suss.edu.sg,

kevinkf.yuen@gmail.com

The story

1. Create a Heron module
2. Record the history how users use this applications
3. Change the user name in the db.

Heron module

- ❖ Create a Heron module (**Heron.py**) with a function calculate the valid inputs of lengths of 3 sides, a, b, c.
- ❖ Create a **testHeron.py** to test the function in the module

Heron.py

```
8 import math
9
10 # create a Heron function
11 def Heron_triangle_area(sides):
12     """
13     calculate triangle area
14     attributes: a iterable set [a, b, c] or (a, b, c)
15                 of non zero and positive value
16     return: triangle area
17     """
18     # Implement the logic to check the validity of code in the future
19     a = sides[0]
20     b = sides[1]
21     c = sides[2]
22     s = (a + b + c) / 2
23     A_square = s * (s - a) * (s - b) * (s - c)
24     A = None
25     if A_square > 0:
26         A = math.sqrt(A_square)
27     else:
28         print(f"The inputs {sides} cannot form a triangle.")
29     return A
```

```
8 # testHeron.py
9 # test the function in Heron module
10 from Heron import Heron_triangle_area
11 test = Heron_triangle_area([4, 13, 15])
12 print("The expected result should be 24: ", test)
13
14 test1 = Heron_triangle_area((0, 1, 15))
15 print("error?: ", test1)
```

```
PS D:\_0SUSS\ANL251Python\MyCode\L4> python testHeron.py
The expected result should be 24: 24.0
The inputs (0, 1, 15) cannot form a triangle.
error?: None
```

Record the history how users use this application

queryAndRecord.py

```
7 from Heron import Heron_triangle_area
8 import datetime
9
10 print("Welcome")
11 user_name = input("What is your name? ")
12 a = float(input("input a positive number for side 1 : "))
13 b = float(input("input a positive number for side 2 : "))
14 c = float(input("input a positive number for side 3 : "))
15 submit_time = str(datetime.datetime.now())
16
17 area = Heron_triangle_area([a, b, c])
18
19 print("The area is ", area)
20 print("Bye!")
21
22 # keep record
23 record = user_name + " " + str(submit_time) + " " \
24         + str([a, b, c]) + " " + str(area) + "\n"
25
26 file = open("log.txt", "a")
27 file.write(record)
28
29 file.close()
```

log.txt

```
1 Kevin 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
2 Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
3 Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
4 Kevin 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
5 Kevin 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
6 John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
7 Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
```

```
PS D:\_0SUSS\ANL251Python\MyCode\L4> python queryAndRecord.py
Welcome
What is your name? Kevin
input a positive number for side 1 : 6
input a positive number for side 2 : 8
input a positive number for side 3 : 10
The area is 24.0
Bye!
```

log.txt has been updated.

New Record is added

```
1 Kevin 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
2 Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
3 Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
4 Kevin 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
5 Kevin 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
6 John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
7 Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
8 Kevin 2019-08-16 16:49:03.078752 [6.0, 8.0, 10.0] 24.0
9
```

New Record

Change the user name in the log.

changeUserName.py

```
1 # assume the name exist
2 print("Change old_name")
3 old_name = input("What is the name to be changed? ")
4 new_name = input("What is the new name? ")
5
6 file1 = open("log.txt", "r")
7 data = file1.readlines()
8
9 temp = []
10 count = 0
11 for line in data:
12     if not line.find(old_name):
13         print("we find one")
14         count += 1
15         temp.append(line.replace(old_name, new_name))
16     else:
17         temp.append(line)
18
19 file2 = open("log.txt", "w")
20 file2.writelines(temp)
21
22 file1.close()
23 file2.close()
24
25 print(f"we change {count} {old_name} to {new_name}")
26 print("bye")
```

```
1 Kevin 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
2 Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
3 Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
4 Kevin 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
5 Kevin 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
6 John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
7 Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
8 Kevin 2019-08-16 16:49:03.078752 [6.0, 8.0, 10.0] 24.0
9
```

```
PS D:\_0SUSS\ANL251Python\MyCode\L4> python changeUserName.py
Change old_name
What is the name to be changed? Kevin
What is the new name? Kevin YUEN
we find one
we find one
we find one
we find one
we change 4 Kevin to Kevin YUEN
bye
PS D:\_0SUSS\ANL251Python\MyCode\L4>
```

```
1 Kevin YUEN 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
2 Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
3 Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
4 Kevin YUEN 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
5 Kevin YUEN 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
6 John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
7 Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
8 Kevin YUEN 2019-08-16 16:49:03.078752 [6.0, 8.0, 10.0] 24.0
9
```