

# hw2\_python\_review

January 26, 2020

## 1 Homework 2

In this homework you will complete a couple of simple exercises in order to show your understanding with Python. If these exercises are challenging or new to you, you may want to reconsider taking the class and/or brush up on your Python skills. For the following exercises you are not allowed to use any Python packages (i.e. Numpy, Pandas, etc.).

### 1.0.1 Please print the output of each question in a new cell below your code

```
[513]: # These lines load the autograder tests. DO NOT change this
from client.api.notebook import Notebook
ok = Notebook('hw02.ok')
ok.auth(inline=True)
score_counter = 0
```

```
=====
Assignment: hw02
OK, version v1.12.5
=====
```

```
Successfully logged in as abrar_syed@berkeley.edu
```

### 1.1 Lists

1.1 Create an empty Python list called 'a' in the cell below.

```
[514]: #your code here
a = []
```

```
[515]: val = ok.grade("q1a")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests
-----
```

```
Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

1.2 Store all values between 1-100 (inclusive) with increments of 3 (i.e. 1, 4, 7...) in 'a'.

```
[516]: #your code here
      for i in range(1,101,3):
          a.append(i)
```

```
[517]: val = ok.grade("q1b")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests
```

```
-----
Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

1.3 Create another list called 'a2' with numbers from 2-46 (inclusive) with increments of 0.5 (i.e. 2, 2.5, 3...).

```
[518]: #your code here
      a2 = []
      i=2.0;
      while(i<=46):
          a2.append(i)
          i+=0.5
```

```
[519]: val = ok.grade("q1c")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests
```

```
-----
Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

1.4 Double every even integer element from list 'a'. Store the results back in 'a'.

```
[520]: #your code here
for i in range(0,len(a)):
    if(a[i]%2==0):
        a[i]=2*a[i]
```

```
[521]: val = ok.grade("q1d")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

~~~~~  
Running tests

-----  
Test summary  
Passed: 1  
Failed: 0  
[ooooooooook] 100.0% passed

1.5 Add all numbers in 'a' except for the 2nd and 21st elements (the 2nd element here means the element at list index 1).

```
[522]: #your code here
sum_a=0
for i in range(0,len(a)):
    if(i!=1 and i!=20):
        sum_a=sum_a+a[i]
print(sum_a)
```

2532

```
[523]: val = ok.grade("q1e")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

~~~~~  
Running tests

-----  
Test summary  
Passed: 1  
Failed: 0  
[ooooooooook] 100.0% passed

1.6 Calculate the mean of 'a'.

```
[524]: #your code here
sum1=0
for i in range(0,len(a)):
    sum1=sum1+a[i]
```

```
mean_a = sum1/len(a)
print(mean_a)
```

76.5

```
[525]: val = ok.grade("q1f")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

~~~~~  
Running tests

-----  
Test summary  
Passed: 1  
Failed: 0  
[ooooooooook] 100.0% passed

1.7 Delete all elements greater than the mean value from list 'a'

```
[526]: for i in range(len(a)):
        if a[i]>mean:
            a.pop(i)
print(a)
```

↳

↳ -----

IndexError Traceback (most recent call↳  
↳last)

<ipython-input-526-e0c9429f294b> in <module>  
1 for i in range(len(a)):  
----> 2 if a[i]>mean:  
3 a.pop(i)  
4 print(a)

IndexError: list index out of range

```
[527]: val = ok.grade("q1g")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

~~~~~  
Running tests

```

-----
> Suite 1 > Case 1

>>> a
[1, 8, 7, 20, 13, 32, 19, 44, 25, 56, 31, 68, 37, 43, 49, 55, 61, 67, 73, 79,
85, 91, 97]

# Error: expected
#      [1, 8, 7, 20, 13, 32, 19, 44, 25, 56, 31, 68, 37, 43, 49, 55, 61, 67, 73]
# but got
#      [1, 8, 7, 20, 13, 32, 19, 44, 25, 56, 31, 68, 37, 43, 49, 55, 61, 67, 73,
79, 85, 91, 97]

Run only this test case with "python3 ok -q q1g --suite 1 --case 1"
-----

Test summary
  Passed: 0
  Failed: 1
[k...] 0.0% passed

```

## 1.2 Strings

2.1 Create an empty list called 'b'.

```

[528]: #your code here
b=[]

[529]: val = ok.grade("q2a")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter

```

~~~~~  
Running tests

```

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed

```

2.2 Store the words in the sentence below as elements into the list 'b'.

'I am so excited about Data-X. It is important to be able to work with data.'

```

[530]: #your code here
string = 'I am so excited about Data-X. It is important to be able to work with
↳data.'

```

```
b= string.split()
print(b)
```

```
['I', 'am', 'so', 'excited', 'about', 'Data-X.', 'It', 'is', 'important', 'to',
'be', 'able', 'to', 'work', 'with', 'data.']
```

```
[531]: val = ok.grade("q2b")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
```

Running tests

```
-----
```

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

2.3 Return the count of the occurrences of the lower-case letter 'e' in the list 'b'.

```
[532]: occurrences=0
for i in b:
    for j in i:
        if j=='e':
            occurrences+=1
print(occurrences)
```

4

```
[533]: val = ok.grade("q2c")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
```

Running tests

```
-----
```

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

2.4 Replace every lower- or upper-case letter 'i' in the list b with a '1'.

```
[534]: for i in range(0,len(b)):
        s=""
        for j in b[i]:
            if j!='i' and j!='I':
```

```

        s+=j
    else:
        s+='1'

    b[i]=s
print(b)

```

```
['1', 'am', 'so', 'excited', 'about', 'Data-X.', 'it', 'is', 'important', 'to',
'be', 'able', 'to', 'work', 'with', 'data.']
```

```
[535]: val = ok.grade("q2d")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```

~~~~~
Running tests

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed

```

2.5 Append the string "This is the end of the first HW." to the list 'b'.

```
[536]: #your code here
t='This is the end of the first HW.'
w=t.split()
b=b+w
print(b)
```

```
['1', 'am', 'so', 'excited', 'about', 'Data-X.', 'it', 'is', 'important', 'to',
'be', 'able', 'to', 'work', 'with', 'data.', 'This', 'is', 'the', 'end', 'of',
'the', 'first', 'HW.']
```

```
[537]: val = ok.grade("q2e")
score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```

~~~~~
Running tests

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed

```

2.6 Print 'b' as ONE string backwards (starting with "WH tsrif...").

```
[538]: print(b)
      b=" ".join(b)
      b_backwards=b[::-1]

['1', 'am', 'so', 'excited', 'about', 'Data-X.', 'it', 'is', 'important', 'to',
'be', 'able', 'to', 'work', 'with', 'data.', 'This', 'is', 'the', 'end', 'of',
'the', 'first', 'HW.']
```

```
[539]: val = ok.grade("q2f")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

## 1.3 Dictionaries

3.1 Put the following in a dictionary called 'codes':

Keys: 1001, 1002, 1003, 1004, 1005

Values: 'Alpha', 'Beta', 'Gamma', 'Delta', 'Tau'

then traverse the dictionary by its keys and change every value to be all lower case.

```
[540]: #your code here
      codes = {1001:'alpha',1002:'beta',1003:'gamma',1004:'delta',1005:'tau'}
```

```
[541]: val = ok.grade("q3a")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

3.2 Delete 'alpha' from the dictionary.



```
[542]: #your code here
      if 1001 in codes:
          del codes[1001]

[543]: val = ok.grade("q3b")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

## 1.4 Sets

4.1 Create a set called 'c' with the all the odd numbers less than 10.

```
[544]: #your code here
      c = {1,3,5,7,9}

[545]: val = ok.grade("q4a")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests

-----

Test summary
  Passed: 1
  Failed: 0
[ooooooooook] 100.0% passed
```

4.2 Create another set called 'd' with elements 2, 5, 10, 30.

```
[546]: #your code here
      d = {2,5,10,30}

[547]: val = ok.grade("q4b")
      score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~
Running tests
```

```
-----  
Test summary  
    Passed: 1  
    Failed: 0  
[ooooooooook] 100.0% passed
```

4.3 Find the union between sets 'c' and 'd' and store this in a new set called 'e'.

```
[548]: #your code here  
       e=c.union(d)  
       print(e)
```

```
{1, 2, 3, 5, 7, 9, 10, 30}
```

```
[549]: val = ok.grade("q4c")  
       score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~  
Running tests
```

```
-----  
Test summary  
    Passed: 1  
    Failed: 0  
[ooooooooook] 100.0% passed
```

4.4 Find the intersection between sets 'c' and 'd' and assign it to a variable **result**.

```
[550]: #your code here  
       result = c.intersection(d)
```

```
[551]: val = ok.grade("q4d")  
       score_counter = score_counter + 1 if val['passed'] == 1 else score_counter
```

```
~~~~~  
Running tests
```

```
-----  
Test summary  
    Passed: 1  
    Failed: 0  
[ooooooooook] 100.0% passed
```

#### 1.4.1 Make sure you save and run the entire file at this point: Kernel -> Restart and Run All

Use the bottom cell for calculating your total expected score for your own reference.

```
[552]: import os
max_score = len([1 for q in os.listdir("tests") if q.startswith('q')])
```

```
[553]: ## For your reference only
print(f'Your score: {score_counter}')
print(f'Max expected score: {max_score}')
```

Your score: 18

Max expected score: 19

#### 1.4.2 Submission

Save this notebook as a pdf file: Ctrl (Cmd) + P for print preview -> Save as PDF and upload the pdf onto the Gradescope assignment.

#### 1.5 Contributors:

Ikhlaq Sidhu: sidhu @ berkeley edu

Arash Nourian: nourian @ berkeley edu

Alexander Fred-Ojala: afo @ berkeley edu

Lillian Dong: lilliandong @ berkeley edu

Ishaan Malhi: ishaan.malhi @ berkeley edu

Sana Iqbal: sanaiqbalsana @ yahoo in