

PYTHON ASSIGNMENT BOOK

MAKE A MOVE TO PYTHON



ASSIGNMENTS

TASK FOUR: FUNCTIONS

Submitted By

Farhana Firoz

Date: May 4, 2020

1. Write a program to reverse a string.

Sample data: "1234abcd"

Expected Output: "dcba4321"

```
def reverse (str1):  
    new=""  
    index=len(str1)  
    while index>0:  
        new+=str1[index-1]  
        index=index-1  
    return new  
print (reverse("1234abcd"))
```

2. Write a function that accepts a string and calculate the number of uppercase letters and lowercase letters.

```
def string_test(s):  
    d={"UPPER_CASE":0, "LOWER_CASE":0}  
    for c in s:  
        if c.isupper():  
            d["UPPER_CASE"]+=1  
        elif c.islower():  
            d["LOWER_CASE"]+=1  
        else:  
            pass  
    print ("Original String : ", s)  
    print ("No. of Upper case characters : ", d["UPPER_CASE"])  
    print ("No. of Lower case Characters : ", d["LOWER_CASE"])  
  
string_test("Coronavirus Will Change the World Permanently")
```

3. Create a function that takes a list and returns a new list with unique elements of the first list.

```
def unique_list(l):  
    x = []  
    for a in l:  
        if a not in x:  
            x.append(a)  
    return x  
print(unique_list([2,3,5,3,7,3,4,5]))
```

4. Write a program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

```
items=[i for i in input().split('-')]
items.sort()
print('-'.join(items))
```

5. Write a program that accepts a sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

Sample input:

Hello world

Practice makes perfect

Expected Output:

HELLO WORLD

PRACTICE MAKES PERFECT

```
lines = []
while True:
    l = input()
    if l:
        lines.append(l.upper())
    else:
        break;

for l in lines:
    print(l)
```

6. Define a function that can receive two integral numbers in string form and compute their sum and print it in console.

```
def printValue(x,y):
    print (int(x)+int(y))

printValue("3","4")
```

7. Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print all strings line by line.

```
def length_of_string(str1, str2):
    if (len(str1) == len(str2)):
        print(str1)
        print(str2)

    elif (len(str1) < len(str2)):
        print(str2)

    else:
        print(str1)
```

```
stri1 = input(str("enter First String: "))
stri2 = input(str("enter Second String: "))

print("\n")

length_of_string(stri1, stri2)
```

8. Define a function which can generate and print a tuple where the value is square of numbers between 1 and 20.

```
def printTuples():
    tuple = [i ** 2 for i in range(1, 21)]
    print(tuple)

printTuples()
```

9. Write a function called showNumbers that takes a parameter called limit. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers.

Example: If the limit is 3, it should print:

0 EVEN

1 ODD

2 EVEN

3 ODDS

```
def showNumbers(limit):
    for i in range(0, limit + 1):
        if i % 2 == 0:
            print(str(i) + ":" + 'EVEN')
        else:
            print(str(i) + ":" + 'ODD')
    limit = int(input("Give me a number. "))

showNumbers(limit)
```

10. Write a program which can filter () to make a list whose elements are even number between 1 and 20 (both included)

```
def even(x):
    return x%2==0
evenNumbers = filter(even, range(1,21))
print(list(evenNumbers))
```

11. Write a program which can map () and filter () to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10]

Hints: Use map () to generate a list.

Use filter () to filter elements of a list

Use lambda to define anonymous functions

```
li = [1,2,3,4,5,6,7,8,9,10]
squaredNumbers = map (lambda x: x**2, li)
print(list(squaredNumbers))
```

12. Write a function to compute 5/0 and use try/except to catch the exceptions

```
try:
    x = 5 / 0
except:
    print("Error dividing by zero")
```

13. Flatten the list [[1,2,3],[4,5],[6,7,8]] into [1,2,3,4,5,6,7,8] using reduce

Goal: Turn [1,2,3,4,5,6,7] to 1234567

```
import operator
from functools import reduce

lists = [[1, 2, 3], [4, 5], [6, 7, 8]]
joinedlist = reduce(operator.add, lists)

print(joinedlist)
```

14. What is the output of the following codes:

(i) def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)

```
Q1 x Occurance x
/Users/onna/PycharmProjects/Practice/venv/bin/python /Users/onna/PycharmProjects/Practice/Q1.py
2

Process finished with exit code 0
```

(ii) def a():

```
    try:
        f(x, 4)
    finally:
        print('after f')
    print('after f?')
a()
```

```
Q1 x Occurance x
/Users/onna/PycharmProjects/Practice/venv/bin/python /Users/onna/PycharmProjects/Practice/Q1.py
after f
Traceback (most recent call last):
  File "/Users/onna/PycharmProjects/Practice/Q1.py", line 7, in <module>
    a()
  File "/Users/onna/PycharmProjects/Practice/Q1.py", line 3, in a
    f(x, 4)
NameError: name 'f' is not defined

Process finished with exit code 1
```

TASK-4

1. Create a list of the 10 elements of four different types of Data Type like int, string, complex and float.

```
x = [2,4,3.6,2.5,3.5j, 5.5j,10, "consultadd", "Training", "Onna"]
print(len(x))
print(type(3.5j))
print(type("Onna"))
print(x[0:5:2])
```

2. Create a list of size 5 and execute the slicing structure.

```
x = [2,4,6,8,10]
print (len(x))
print (x[0:5:2])
```

3. Create a list of given structure and run

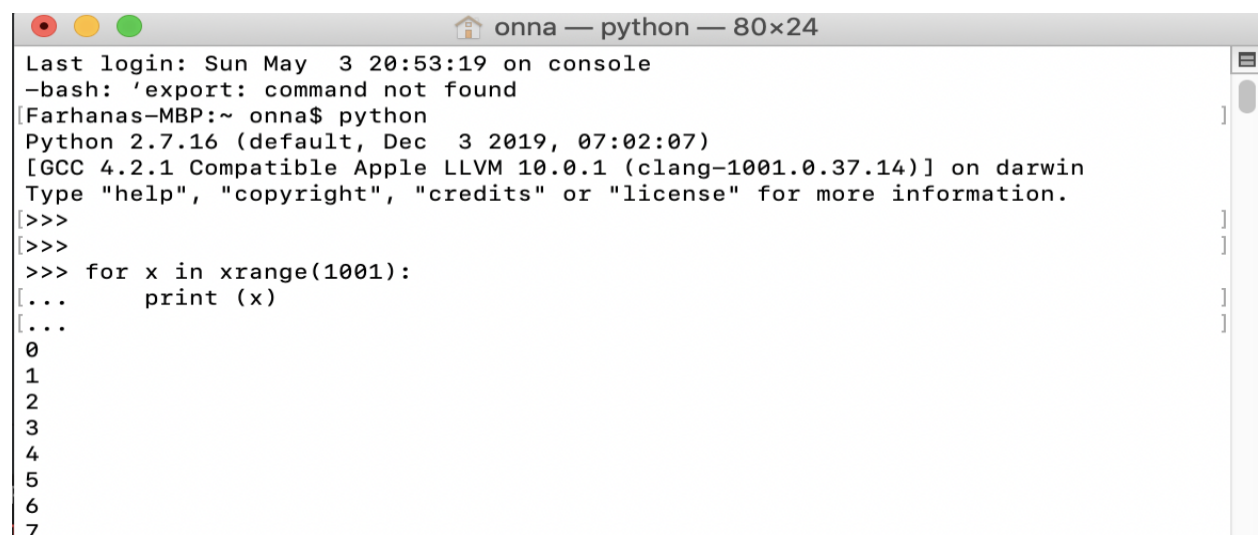
```
x=[100,200,300,400,500,[1,2,3,4,5],[10,20,30,40,50],6,7,8,9],600,700,800]
```

- Access list [1, 2, 3, 4]
- Access list [600, 700]
- Access list [100, 300, 500, 600, 800]
- Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]]
- Access list [10]
- Access list []

```
• x=[100,200,300,400,500,[1,2,3,4,5],[10,20,30,40,50],6,7,8,9],600,700,800]
  Access list [1, 2, 3, 4]
  Access list [600, 700]
  Access list [100, 300, 500, 600, 800]
  Access list [[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300, 200, 100]]
  Access list [10]
  Access list [ ]
```

4. Create a list of thousand number using range and xrange and see the difference between each other.

```
for x in range (1001):
    print (x)
for x in range (0,1000,4):
    print (x)
```



```
onna — python — 80x24
Last login: Sun May  3 20:53:19 on console
-bash: 'export: command not found
[Farhanas-MBP:~ onna$ python
Python 2.7.16 (default, Dec  3 2019, 07:02:07)
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>>
[>>>
>>> for x in xrange(1001):
[...     print (x)
[...
0
1
2
3
4
5
6
7
```

5. How Tuple is beneficial as compare to the list?

In general, Tuples and lists have the same uses. Immutable data types have many benefits, mostly about concurrency issues. So, when you have lists that are not volatile in nature and you need to guarantee that no consumer is altering it, you may use a tuple. Tuples are used whenever you want to return multiple results from a function and it's faster than list. Since they're immutable, they can be used as keys for a dictionary (lists can't).

6. Write a program in Python to iterate through the list of numbers in the range of 1,100 and print the number which is divisible by 3 and a multiple of 2.

```
lower = int(input("Enter lower range limit:"))
upper = int(input("Enter upper range limit:"))
for i in range(lower, upper+1):
    if((i%3==0) & (i*2==0)):
        print(i)
```

7. Write a program in Python to reverse a string and print only the vowel alphabet if exist in the string with their index.

```
str="Python"
stringlength=len(str)
slicedString=str[stringlength::-1]
print (slicedString)

def Check_Vow(string, vowels):
    python = [each for each in string if each in vowels]
    print(len(python))
    print(python)
string = "python"
vowels = "AaEeliOoUu"
Check_Vow(string, vowels);
```

8. Write a program in Python to iterate through the string “hello my name is abcde” and print the string which has even length of word.

```
def printWords(s):
    s = s.split(' ')

    for word in s:

        if len(word) % 2 == 0:
            print(word)

s = "hello my name is abcde"
printWords(s)
```

9. Write a program in python to print the pair of numbers whose sum is equal to result number that is let's say 8.

`x=[1,2,3,4,5,6,7,8,9,-1]`

```
class py_solution:
    def twoSum(self, nums, target):
        lookup = {}
        for i, num in enumerate(nums):
            if target - num in lookup:
                return (lookup[target - num], i)
            lookup[num] = i
print("index1=%d, index2=%d" % py_solution().twoSum([1,2,3,4,5,6,7,8,9,-1],8))
```

10. Write a program in Python to complete the following task:

- Create two different list as in even_list and odd_list
- Ask user to enter the number in the range of 1,50 and make sure if the entered number is even append it to the even_list and if the entered number is odd append it to the odd list.
- Keep that in mind you can only add 5 items in each list
- Make sure once you entered the total 5 elements calculate the sum of the list and return the maximum out of the list.

```
• def splitevenodd(A):
    evenlist = []
    oddlist = []
    for i in range(51):
        evenlist.append(i)
    else:
        oddlist.append(i)
    print("Even lists:", evenlist)
    print("Odd lists:", oddlist)

A = list()
n = int(input("Enter the size of the First List ::"))
print("Enter the Element of First List ::")
for i in range(int(51)):
    k = int(input(""))
    A.append(k)
splitevenodd(A)
```

11. Write a program to find out the occurrence of a specific word from an alphanumeric statement. Example: 12abcbacbaba344ab

Output: a=5 b=5 c=2 make sure you should avoid the numbers in your logic

```
def check(str1):
    new_dict={}
    for i in str1:
        if i.isalpha():
            if i in new_dict:
                new_dict[i]+=1
            else:
```

```
        new_dict[i]=1
    return new_dict
print (check("12abcbacbaba344ab"))
```

12. Generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

```
x = (1,2,3,4,5,6,7,8,9,10)
y = tuple (i for i in x if i%2 == 0)
print(y)
```