Python Training  
WEEKEND ACTIVITY ON FUNCTIONS

By: Utkarsh Dandwate

**Q1. Write a program to reverse a string.**

**Sample data: “1234abcd”**

**Expected Output: “dcba4321”**

**🡪** def reverse\_string():  
 x = input("Please enter a string input: ")  
 print("The reverse of given string is: ", x[::-1])  
  
reverse\_string()

**Q2. Write a function that accepts a string and calculate the number of uppercase letters**

**and lowercase letters.**

**Expected Output:**

**No. of Upper case characters : 3**

**No. of Lower case Characters : 12**

**🡪**def upperlowerchar():  
 x = input("Please enter a string: ")  
 upper = 0  
 lower = 0  
 for i in range(len(x)):  
 if x[i].isupper():  
 upper += 1  
 else:  
 lower += 1  
 print("No. of uppercase characters: ", upper, "\n"  
 "No. of lowercase characters: ", lower)  
  
upperlowerchar()

**Q3. Create a function that takes a list and returns a new list with unique elements of the**

**first list.**

**🡪** def unique\_elements():  
 x = input("Enter a list of numbers separated by comas: ")  
 x = x.split(",")  
 y = []  
 for i in range(len(x)):  
 if x[i] not in y:  
 y.append(x[i])  
 print("Unique elements of the entered list are: ", y)  
  
  
unique\_elements()

**Q4.** **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.**

**🡪**def alphasort():  
 x = input("Enter hyphen separated sequence of words: \n ")  
 y = x.split("-")  
 y.sort()  
  
 print("Sequence of words after sorting alphabetically is : ", '-'.join(y) )  
alphasort()

**Q5. 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**

**🡪**def q5function():  
 print("Enter sequence of lines: ")  
 entire\_input = []  
 while True:  
 x = input()  
 if x:  
 entire\_input.append(x)  
 else:  
 break  
 multiline\_input = "\n".join(entire\_input)  
   
 print(multiline\_input.upper())  
  
  
  
q5function()

**Q6.** **Define a function that can receive two integral numbers in string form and compute**

**their sum and print it in console.**

**🡪**def addition():  
 x = eval(input("Enter first number:\n"))  
 y = eval(input("Enter second number: \n"))  
 print("The sum of two numbers is: ", x + y )  
  
addition()

**Q7. 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 lenghtofstrings():  
 x = input("Enter first string: ")  
 y = input("Enter second string: ")  
 if len(x) == len(y):  
 print(x + '\n'+ y)  
 if len(x) > len(y):  
 print(x)  
 elif len(y) > len(x):  
 print(y)  
  
lenghtofstrings()

**Q8. Define a function which can generate and print a tuple where the value are square of**

**numbers between 1 and 20.**

**🡪**def tuple1to20():  
 x = []  
 for i in range(1,21):  
 x.append(i\*\*2)  
 y = tuple(x)  
 print(y)  
  
tuple1to20()

**Q9. 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 ODD**

**🡪**def showNumbers(limit):  
 for i in range((limit+1)):  
 if i%2 == 0:  
 print( i , " EVEN")  
 else:  
 print(i , " ODD")  
  
  
showNumbers(3)

**Q10. Write a program which can filter() to make a list whose elements are even number**

**between 1 and 20 ( both included)**

**🡪**result = filter(lambda x : x % 2 == 0 , range(1,21))  
print(list(result)

**Q11. 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**

**🡪**result\_1 = list(map(lambda x : x\*x, range(1,11)))  
final\_result = list(filter(lambda x : x%2 == 0, result\_1))  
print(final\_result)

**Q12. Write a function to compute 5/0 and use try/except to catch the exceptions.**

**🡪** def division\_by\_zero():  
 try:  
 x = eval(input("Enter first number: "))  
 y = eval(input("Enter second number: "))  
 result = x / y  
 print(result)  
 except:  
 print("Cannot divide by zero, please enter a non-zero second number. ")  
  
division\_by\_zero()

**Q13. 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 functools  
result = functools.reduce(lambda x , y : x + y , [[1,2,3],[4,5],[6,7,8]] )  
print(result)

**Q14.What is the output of the following codes:**

**(i) def foo():**

**try:**

**return 1**

**finally:**

**return 2**

**k = foo()**

**print(k)**

**🡪 2**

**(ii) def a():**

**try:**

**f(x, 4)**

**finally:**

**print('after f')**

**print('after f?')**

**a()**

**🡪 after f**