Python course materials

# Functions and Methods Homework Solutions

**Write a function that computes the volume of a sphere given its radius.**

def vol(rad):  
 return (4/3)\*(3.14)\*(rad\*\*3)

# Check  
vol(2)

33.49333333333333

**Write a function that checks whether a number is in a given range (inclusive of high and low)**

def ran\_check(num,low,high):  
 #Check if num is between low and high (including low and high)  
 if num in range(low,high+1):  
 print('{} is in the range between {} and {}'.format(num,low,high))  
 else:  
 print('The number is outside the range.')

# Check  
ran\_check(5,2,7)

5 is in the range between 2 and 7

If you only wanted to return a boolean:

def ran\_bool(num,low,high):  
 return num in range(low,high+1)

ran\_bool(3,1,10)

True

**Write a Python function that accepts a string and calculates the number of upper case letters and lower case letters.**

Sample String : 'Hello Mr. Rogers, how are you this fine Tuesday?'  
Expected Output :   
No. of Upper case characters : 4  
No. of Lower case Characters : 33

If you feel ambitious, explore the Collections module to solve this problem!

def up\_low(s):  
 d={"upper":0, "lower":0}  
 for c in s:  
 if c.isupper():  
 d["upper"]+=1  
 elif c.islower():  
 d["lower"]+=1  
 else:  
 pass  
 print("Original String : ", s)  
 print("No. of Upper case characters : ", d["upper"])  
 print("No. of Lower case Characters : ", d["lower"])

s = 'Hello Mr. Rogers, how are you this fine Tuesday?'  
up\_low(s)

Original String : Hello Mr. Rogers, how are you this fine Tuesday?  
No. of Upper case characters : 4  
No. of Lower case Characters : 33

**Write a Python function that takes a list and returns a new list with unique elements of the first list.**

Sample List : [1,1,1,1,2,2,3,3,3,3,4,5]  
Unique List : [1, 2, 3, 4, 5]

def unique\_list(lst):  
 # Also possible to use list(set())  
 x = []  
 for a in lst:  
 if a not in x:  
 x.append(a)  
 return x

unique\_list([1,1,1,1,2,2,3,3,3,3,4,5])

[1, 2, 3, 4, 5]

**Write a Python function to multiply all the numbers in a list.**

Sample List : [1, 2, 3, -4]  
Expected Output : -24

def multiply(numbers):  
 total = 1  
 for x in numbers:  
 total \*= x  
 return total

multiply([1,2,3,-4])

-24

**Write a Python function that checks whether a word or phrase is palindrome or not.**

Note: A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam,kayak,racecar, or a phrase “nurses run”. Hint: You may want to check out the .replace() method in a string to help out with dealing with spaces. Also google search how to reverse a string in Python, there are some clever ways to do it with slicing notation.

def palindrome(s):  
   
 s = s.replace(' ','') # This replaces all spaces ' ' with no space ''. (Fixes issues with strings that have spaces)  
 return s == s[::-1] # Check through slicing

palindrome('nurses run')

True

palindrome('abcba')

True

#### Hard:

**Write a Python function to check whether a string is pangram or not. (Assume the string passed in does not have any punctuation)**

Note : Pangrams are words or sentences containing every letter of the alphabet at least once.  
For example : "The quick brown fox jumps over the lazy dog"

Hint: You may want to use .replace() method to get rid of spaces.

Hint: Look at the [string module](https://stackoverflow.com/questions/16060899/alphabet-range-in-python)

Hint: In case you want to use [set comparisons](https://medium.com/better-programming/a-visual-guide-to-set-comparisons-in-python-6ab7edb9ec41)

import string  
  
def ispangram(str1, alphabet=string.ascii\_lowercase):   
 # Create a set of the alphabet  
 alphaset = set(alphabet)   
   
 # Remove spaces from str1  
 str1 = str1.replace(" ",'')  
   
 # Lowercase all strings in the passed in string  
 # Recall we assume no punctuation   
 str1 = str1.lower()  
   
 # Grab all unique letters in the string as a set  
 str1 = set(str1)  
   
 # Now check that the alpahbet set is same as string set  
 return str1 == alphaset

ispangram("The quick brown fox jumps over the lazy dog")

True

string.ascii\_lowercase

'abcdefghijklmnopqrstuvwxyz'