Python les-materialen

# Functies en methoden Huiswerk met oplossingen

Vul de volgende vragen in: \_\_\_\_ **Schrijf een functie die het volume van een bol (sphere) berekent op basis van zijn straal (radius).**

Het volume van een bol wordt gegeven als

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

# Check  
vol(2)

33.49333333333333

**Schrijf een functie die controleert of een getal binnen een bepaald bereik valt (inclusief hoog en laag)**

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.')

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

5 is in the range between 2 and 7

Als je alleen een boolean wilt retourneren:

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

ran\_bool(3,1,10)

True

**Schrijf een Python-functie die een string accepteert en het aantal hoofdletters en kleine letters berekent.**

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

HINT: Twee string-methoden die nuttig kunnen zijn: **.isupper()** en **.islower()**

Als je ambitieus bent, verken dan de module Collecties om dit probleem op te lossen!

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

**Schrijf een Python-functie die een lijst nodig heeft en een nieuwe lijst retourneert met unieke elementen van de eerste lijst.**

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]

**Schrijf een Python-functie om alle getallen in een lijst te vermenigvuldigen (multiply).**

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

**Schrijf een Python-functie die controleert of een woord of zin palindroom is of niet.**

Opmerking: een palindroom is een woord, zin of reeks die achterstevoren hetzelfde leest als vooruit, bijvoorbeeld “madam”, “kayak”, “racecar” of een zinsnede “nurses run”. Hint: misschien wil je de methode .replace() in een string bekijken om te helpen bij het omgaan met spaties. Zoek ook op Google hoe u een string in Python kunt omkeren, er zijn enkele slimme manieren om dit te doen met slicing-notatie.

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

#### Moeilijk:

**Schrijf een Python-functie om te controleren of een string pangram is of niet. (Stel dat de ingevoerde tekenreeks geen leestekens/punctuation heeft)**

Let op: Pangrammen zijn woorden of zinnen die elke letter van het alfabet minstens één keer bevatten.  
 Bijvoorbeeld: "The quick brown fox jumps over the lazy dog"

Hint: misschien wilt u de methode .replace() gebruiken om spaties te verwijderen.

Hint: kijk naar de [string-module](https://stackoverflow.com/questions/16060899/alphabet-range-in-python)

Hint: als je [set-vergelijkingen](https://medium.com/better-programming/a-visual-guide-to-set-comparisons-in-python-6ab7edb9ec41) wilt gebruiken

import string  
  
def is\_pangram(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 alphabet set is same as string set  
 return str1 == alphaset

is\_pangram("The quick brown fox jumps over the lazy dog")

True

string.ascii\_lowercase

'abcdefghijklmnopqrstuvwxyz'