09-Functions and Methods Homework - Solutions

July 30, 2019

1 Functions and Methods Homework Solutions

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

[1]:
```
def vol(rad):
    return (4/3)*(3.14)*(rad**3)
```

[2]:
```
# Check
vol(2)
```

[2]:
```
33.49333333333333
```

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

[3]:
```
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.')
```

[4]:
```
# Check
ran_check(5,2,7)
```

```
5 is in the range between 2 and 7
```

If you only wanted to return a boolean:

[5]:
```
def ran_bool(num,low,high):
    return num in range(low,high+1)
```

[6]:
```
ran_bool(3,1,10)
```

[6]:
```
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!

[7]:

```python
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"])
```

[8]:

```python
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]
```

[9]:

```python
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
```

[10]:

```python
unique_list([1,1,1,1,2,2,3,3,3,3,4,5])
```

[10]:

```
[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
```

[11]:

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

[12]:

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

[12]:

```
-24
```

Write a Python function that checks whether a passed string is palindrome or not.

Note: A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

[13]:

```
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
```

[14]:

```
palindrome('nurses run')
```

[14]:

```
True
```

[15]:

```
palindrome('abcba')
```

[15]:

```
True
```

Hard:

Write a Python function to check whether a string is pangram or not.

```
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: Look at the string module

[16]:

```
import string

def ispangram(str1, alphabet=string.ascii_lowercase):
    alphaset = set(alphabet)
    return alphaset <= set(str1.lower())
```

[17]:
```
ispangram("The quick brown fox jumps over the lazy dog")
```

[17]:
```
True
```

[18]:
```
string.ascii_lowercase
```

[18]:
```
'abcdefghijklmnopqrstuvwxyz'
```