Functions and Methods Homework

Complete the following questions:

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

In [2]:
```
def vol_sphere(rad):
    vol = ((float(4))/3)*3.14*(rad**3)
    return vol

print vol_sphere(0.5)
```

```
0.523333333333
```

In [3]:
```
def vol(rad):
    vol = (4.0/3)*3.14*(rad**3)
    return vol

vol(3)
```

Out[3]:
```
113.03999999999999
```

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

In [6]:
```
def range_check(low,high,num):
    if num>=low and num<=high:
        print 'in the range'
    else:
        print 'out of range'

range_check(1,66,55)
```

```
in the range
```

In [5]:
```
def ran_check(num,low,high):
    if low<=num<=high:
        print num, ' is in the range'
    else:
        print num, ' is not in the range'

ran_check(-3,-1,100)
```

```
-3  is not in the range
```

If you only wanted to return a boolean:

In [7]:
```
def ran_check(num,low,high):
    if low<=num<=high:
        return True
    else:
        return False

ran_check(-3,-1,100)
```

Out[7]: False

In [8]:
```
def ran_bool(num,low,high):
    if low<=num<=high:
        return True
    else:
        return False
```

In [10]:
```
ran_bool(-3,1,10)
```

Out[10]: False

Write a Python function that accepts a string and calculate 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!

In [12]:
```
len('Sample String')
```

Out[12]: 13

In [13]:
```
def letter_count(s):
    print 'Sample String: %s' %(s)
    upper = 0
    lower = 0
    for n in range(0, len(s)):
        if s[n].isupper():
            upper += 1
        elif s[n].islower():
            lower += 1
    print 'No. of Upper case characters: %s'%(upper)
    print 'No. of Lower case characters: %s'%(lower)

letter_count('Hello Mr. Rogers, how are you this fine Tuesday?')
```

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

In [23]:
```
def up_low(s):
    print 'Sample String: %s' %(s)
    upp = 0
    low = 0
    for letter in s:
        if letter.isupper():
            upp +=1
        elif letter.islower():
            low +=1
    print 'No. of Upper case characters: ', upp
    print 'No. of Lower case characters: ', low

up_low('Hello Mr. Rogers, how are you this fine Tuesday?')
```

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

In [14]:
```
'a b c'.upper()
```

Out[14]:
```
'A B C'
```

In [17]:
```
'A'.isupper()
```

Out[17]:
```
True
```

In [18]:
```
def up_low(s):
    print 'Sample String: %s' %(s)
    for letter in s:
        print letter

up_low('Hello Mr. Rogers, how are you this fine Tuesday?')
```

```
Sample String: Hello Mr. Rogers, how are you this fine Tuesday?
H
e
l
l
o

M
r
.

R
o
g
e
r
s
,

h
o
w

a
r
e

y
o
u

t
h
i
s

f
i
n
e

T
u
e
s
d
a
y
?
```

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]
```

In [16]:
```
def unique_set(l):
    set = []
    for element in l:
        if element not in set:
            set.append(element)
    print set

unique_set([1,1,1,1,2,2,3,3,3,3,4,5])
```

```
[1, 2, 3, 4, 5]
```

In [24]:
```
def unique_list(l):
    return set(l)
```

In [25]:
```
unique_list([1,1,1,1,2,2,3,3,3,3,4,5])
```

Out[25]:
```
{1, 2, 3, 4, 5}
```

In [27]:
```
def unique_list(l):
    unique_list =[]
    for n in l:
        if n not in unique_list:
            unique_list.append(n)
    return unique_list
```

In [29]:
```
unique_list([1,1,1,1,2,2,3,3,3,3,4,5])
```

Out[29]:
```
[1, 2, 3, 4, 5]
```

In [28]:
```
unique_list('I am a great man who does great things')
```

Out[28]:
```
['I', ' ', 'a', 'm', 'g', 'r', 'e', 't', 'n', 'w', 'h', 'o', 'd', 's', 'i']
```

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

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

In [17]:
```
def multiply(s):
    product =1
    for n in s:
        product *= n
    return product

multiply([50,66,789,9876])
```

Out[17]:
```
25714141200L
```

In [30]:
```
def multiply(numbers):
    product =1
    for n in numbers:
        product = n*product
    return product
```

In [33]:
```
multiply([1,2,3,-106])
```

Out[33]:
```
-636
```

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.

In [21]:
```
def panlindrome(s):
    s = s.replace(' ','')
    if s == s[::-1]:
        print '%s is a panlindrome'%s
    else:
        print '%s is not a panlindrome'%s

panlindrome('wang is wang')
```

```
wangiswang is not a panlindrome
```

In [38]:
```
def palindrome(s):
    s=s.replace(' ', '')
    if s==s[::-1]:
        return True
    else:
        return False
```

In [39]:
```
palindrome('helleh')
```

Out[39]:
```
True
```

In [40]:
```
palindrome('nurses run')
```

Out[40]:
```
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

In [24]:
```
import string

def pangram_check(s, alphabet = string.ascii_lowercase):
    s = s.replace(' ','')
    s = s.lower()
    for n in alphabet:
        if n in s:
            return True
        else:
            return False

pangram_check('The quick brown fox jumps over the lazy dog')
```

Out[24]:
```
True
```

In [61]:
```
import string

def ispangram(str1, alphabet=string.ascii_lowercase):
    str1= str1.replace(' ','')
    str1=str1.lower()
    str1= set(str1)
    alphabet =set(string.ascii_lowercase)
    if str1==alphabet:
        return True
    else:
        return False
```

In [63]:
```
ispangram("The quick rown fox jumps over the lazy dog")
```

Out[63]:
```
False
```

In [64]:
```
ispangram("The quick brown fox jumps over the lazy dog")
```

Out[64]:
```
True
```

In [23]:
```
string.ascii_lowercase
```

Out[23]:
```
'abcdefghijklmnopqrstuvwxyz'
```

In [65]:
```
s=[5,3,4,2,1]
set(s)
```

Out[65]:
```
{1, 2, 3, 4, 5}
```

In [66]:
```
s='a c a b c d'
set(s)
```

Out[66]:
```
{' ', 'a', 'b', 'c', 'd'}
```

####Great Job!