

# Functions and Methods Homework Solutions

---

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

```
In [25]: def vol(rad):  
         return (4.0/3)*(3.14)*(rad**3)
```

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

```
In [7]: 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 " %s is in the range" %str(num)  
        else :  
            print "The number is outside the range."
```

If you only wanted to return a boolean:

```
In [8]: def ran_bool(num,low,high):  
        return num in range(low,high+1)
```

```
In [9]: ran_bool(3,1,10)
```

```
Out[9]: True
```

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 [11]: 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"]
```

```
In [12]: 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]

```
In [13]: def unique_list(l):
          # Also possible to use list(set())
          x = []
          for a in l:
              if a not in x:
                  x.append(a)
          return x
```

```
In [14]: unique_list([1,1,1,1,2,2,3,3,3,3,4,5])
```

```
Out[14]: [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

```
In [4]: def multiply(numbers):  
        total = 1  
        for x in numbers:  
            total *= x  
        return total
```

```
In [7]: multiply([1,2,3,-4])
```

```
Out[7]: -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.

```
In [5]: 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
```

```
In [7]: palindrome('nurses run')
```

```
Out[7]: True
```

```
In [8]: palindrome('abcba')
```

```
Out[8]: 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 [21]: import string  
  
def ispangram(str1, alphabet=string.ascii_lowercase):  
    alphaset = set(alphabet)  
    return alphaset <= set(str1.lower())
```

```
In [22]: ispangram("The quick brown fox jumps over the lazy dog")
```

```
Out[22]: True
```

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

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