**Functions**

* Default arguments
  + at the time of function definition we will give input arguments
  + Example for default arguments in range() function it is a bulitin function
  + Range function has default arguments through function call which are declared at the function definition default arguments are used for performing operation
  + Example of default arguments:
    - def greet(name, msg = "Good morning!"):
      * + print("Hello",name + ', ' + msg)
    - greet("Kate")
    - greet("Bruce","How do you do?")
    - output:
    - Hello Kate, Good morning!
    - Hello Bruce, How do you do?
* Variable arguments:
  + Sometimes, we do not know in advance the number of arguments that will be passed into a function.Python allows us to handle this kind of situation through function calls with arbitrary number of arguments.
  + Example:
  + def avg\_n(a,b,\*args):
  + sum=a+b
  + count=0
  + for i in list(args):
  + for j in i:
  + sum=sum+j
  + count=count+1
  + print(sum/(count+2))
  + avg\_n(10,20,[1,2,3,4])
  + output:
  + 6.666666666666667
* Keyword arguments:
  + The double asterisk form of \*\*kwargs is used to pass a keyworded, variable-length argument dictionary to a function
  + example:
  + def hi(\*\*kwargs):2
  + for i,j in kwargs.items():
  + print("Hi i am {} from {}".format(i,j))
  + hi(superman="usa",batman="pk",ironman="uk")
  + ouput:
  + Hi i am superman from usa
  + Hi i am batman from pk
  + Hi i am ironman from uk
* Modules
  + Bulitin Modules:
    - some of the bulitin modules are
      * sys:
        + This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.
      * Random:
        + integers

uniform within range

* + - * + sequence

pick random element

pick random sample

pick weighted random sample

generate random permutation

* + - * os :
        + Programs that import and use 'os' stand a better chance of being portable between different platforms. Of course, they must then only use functions that are defined by all platforms (e.g., unlink and opendir), and leave all pathname manipulation to os.path (e.g., split and join).
      * Time:
        + There are two standard representations of time. One is the number of seconds since the Epoch, in UTC (a.k.a. GMT). It may be an integer or a floating point number (to represent fractions of seconds). The Epoch is system-defined; on Unix, it is generally January 1st, 1970. The actual value can be retrieved by calling gmtime(0)
      * calendar:
        + Note when comparing these calendars to the ones printed by cal(1): By default, these calendars have Monday as the first day of the week, and Sunday as the last (the European convention). Use setfirstweekday() to set the first day of the week (0=Monday, 6=Sunday)
      * math:
        + This module is always available. It provides access to the mathematical functions defined by the C standard
  + User defined Modules:
    - Modules refer to a file containing Python statements and definitions.
    - We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code.
    - We can define our most used functions in a module and import it, instead of copying their definitions into different programs.
    - We use the import keyword to do this. To import our previously defined module
    - import math as m
      * we can the rename the math module with m
      * We can import specific names from a module without importing the module as a whole. By using ‘from’ keyword
      * example from math import pi.