5 Functions

In mathematics, a function specifies a rule of computation. So is the case with python; only the syntax of *defining* a function is different. As an example, conider the computation of compound interest done earlier. Mathematically it involves the computation of the total amount a in terms of the principal p, interest rate r and the number of years n using the formula

$$a(p, r, n) = p\left(1 + \frac{r}{100}\right)^n$$

and to be practically we round it (half up) to the nearest integer. We define this (in fact any function) in Python in two lines of code. To see this, start Python (console mode) and type at the >>> prompt, the name we want to give t the function; it can be a single letter or any combintation of letters as in:

```
>>> def compound(p,r,n):
```

On pressing Enter, we get the secondary prompt ..., since the definition is not complete. to complete it, we must state what we want the function to compute and *return*:

```
>>> def compound(p,r,n):
...  return(round(p*(1+r/100)**n))
```

(Be careful about the indentation). On pressing Enter twice, we get the primary prompt >>> back. Now continue something like

```
>>> def compound(p,r,n):
...  return(round(p*(1+r/100)**n))
...
>>> compound(1000,7,3)
```

and hitting Enter gives the output 1225. And we can continue to compute various amounts using the compound function with different sets of nnumbers.

But then once you quit the current Python session, the definition vanishes, so that the next time you start Python and want to use this function, you will have to define it all over again! One way to overcome this is to make a file myfunctions.py (the name can be anything, but the extension must be .py) containing this function as

```
# This is a function to compute the compound interest

def compound(p,r,n):
 return(round(p*(1+r/100)**n))
```

Now if we want to use this function in a Python console, first type

```
>>> from myfunctions import compound
```

(Does this remind you of something done earlier?) Then you can continue with:

```
>>> from myfunctions import compound
>>> compound(1000,7,5)
1403
>>>
```

We can add morer definitions in myfunctions.py (or whatever you choose to call this file) and import any of these as required

There's another use of functions in Python. To see this, first consider the problem of computing factorials. This can be done with a program using a simple iteration:

```
# A program to compute the factorial of a natural number

print()

n = int(input("Enter Number : "))

print()

for i in range(1, n):
 n = n * i
print(n)

print()
```

(The empty `print()` commands just add some blank lines in the output, making it easier to read.)

Now consider the problem of approximating the number e. It becomes easy if we defime the factorial with in our code:

```
# A program to approximate e

def factorial(n):
 for i in range(1,n):
  n = n * i
 return(n)

print()

n=int(input("Number of terms :"))

print()

s=1
for i in range(1,n+1):
 s=s+1/factorial(i)
print(s)

print()
```

(This is an example what in programming parlance is called a *subroutine*, which simply means using a piece of code repeatedly, without explicitly writingit out in full everytime.) It would be a nice exxercise to think about doing this without functions and using only iterations.