Introduction to computing and data handling 2023

2nd class Loops, conditions, functions, *numpy*, plotting. 31st Aug

A *function* is small code that performs some operations. Functions often have one or more *arguments*. And it also *returns* a result.
We have already seen some built-in functions (for example, *type*).

Python has modules or libraries that provide additional functions. For example, *math* is a module providing familiar mathematical functions. Before one can use the module, it has to be imported.

>>>import math
This statement creates a **module object** named math.
>>>print(math) #and see what it results.
To access functions inside the math module, one has to use the dot notation.
>>> x= 1.5 #in radians
>>>print(math.sin(x))

Other ways to import modules
>>>import math as m
*>>>from math import **
Find out how these methods are different from the first one. What is a disadvantage of the last method? Usually, the 2nd method is most convenient.

Defining your own functions

A function definition
(i) specifies the name of the function,
(ii) should have the necessary arguments,
(iii) and should have a return statement if required
(iv) The set of statements constituting the function should have indentation. Rather indentation defines the set of statements as a block.

```
>>>def  this_function(t):
   '''
   This function accepts a string as argument and prints its value.
   '''
>>>    print(t)

#The main program starts here
>>>this_function('cat')

>>>def  this_function(x):
   '''
   This function accepts a float as argument, converts it to a string, and returns.
   '''
>>>    t=str(x)
>>>    return t

#The main program starts here
>>>this_function(2.5)
```

Loops

Loops are used for iterative calculations. Two important loops are *for* and while. Looping can be done over lists, tuples, and arrays. Let us start with a *numpy* array and see how looping works.

```
>>>import numpy as np #imports the numpy libraries.
```

Check functions *arange, linspace,* and *logspace*. See what are the most important arguments of these functions. What are the default values.

Conditions and *if* statements.

All usual logical operations are supported.

Some examples.

```
>>>x=3.0
>>>if x<2.0:
>>>     print(x)


>>>if x<=2.0:
>>>     print(x)
>>>else:
>>>     print(x*x)

>>>if x>5: print('x is greater than 5')
```

Exercise

Use *numpy* arrays, *for* loops, *if* conditions and obtain the blackbody intensity for T=5600K, from wavelengths ranging from 2000 to 8000 Angstrom. Obtain the same corresponding Rayleigh-Jeans and Wein's limits.

Do the same code but this time using two functions defined on your own. One for Planck function and the other for the approximations (Rayleigh Jeans and Wein's).

Plotting

The python plotting library most commonly used is *matplotlib.* Use this and plot the blackbody curve.

```
>>>import matplotlib.pyplot as plt
>>>plt.plot(lamda, I_lambda)
```