**2.10 Python Data types**

Before proceeding with computational work, the computer must know the exact nature of the data. This is to save time and memory space. Depends on the nature of the operation, python can deal with the following data types.

1 Numeric

2 Sequence

3 Set

4 Dictionary

The numeric data types are integer, float, long integer and complex. Integer type means a plain integer without a decimal part. The length of the integer depends on the microprocessor. Python language cannot deal with an integer above this limit. To get integers of more length we can use long integer type. Float represents the decimal number with 32-bit accuracy. The numbers higher than this will automatically shift to exponential form. Complex numbers can deal with a real part and an imaginary part.

String, Tuple and List are the data types coming under the category sequences. We will discuss Sequence, Set and Dictionary later.

**Rules of type conversion**

1. During addition and subtraction, if both are integer, the result will be an integer. If anyone or both is a float, the result will be a float.
2. During division or multiplication, if both operands are integer, the result will be a truncated integer.
3. If anyone operand is a long integer, the result will be a long integer.
4. If anyone operand is a float and the other is a float or integer, the result will be a float.

During the programme, we must give extra care to deal with the data type. For example, let x=1, y=2, z= x/y. The value of z becomes0 which is undesirable. This is the truncated integer of 0.5. To solve such situations, we must convert one data type to another depends on the situations. The instructions for this are known as ***data type casting instructions***.

*z=float(x)/y*

This is to instruct the computer to convert x to decimal accuracy. Then calculate x/y. The answer is to be stored in z as a float.

*z=float(x/y)* will yield another result.

To convert the quantity ‘per’ to a float, we can use - per=float (per)

Similarly, the instruction, no=int(no)can be used to convert a float to an integer. To get full digit accuracy during the time data entry, we can ask the console to read the number as a long integer. This can be done by any one of the following instructions. Go through the following different instructions.

*rad=long(input(“Enter the radius of an atom”)*

*rad=int(input(“Enter the radius of an atom” )*

*rad=float(input(“Enter the radius of an atom”)*

**List of type casting instructions**

1. *y=float(x):* Convert an integer to float
2. *y=int(x):* Convert a float to integer by truncation.
3. *y=long(x):* Convert an integer or float to a long integer.
4. *z=complex(x,y):* Create a complex number with x as real part and y as imaginary part If *y* is omitted, it defaults to zero

*x=2*

*y=3*

*z=complex(x,y)*

*Then z=(2+3j)*

1. *str(x):* Convert a number to a string. The output will not support any mathematical functions.

*x=2.75*

*y=str(x)*

Then value stored for y will be=’2.75’. It does not have any property of a number. The value stored for y +y will be=’2.752.75’ due to string concatenation

The instruction *‘print (type(x))’* is to know the data type of a quantity represented by x. Go through the following python programme to demonstrate more string activities.