DATA TYPES IN PYTHON

Every value belongs to a specific data type in python. Data type identifies the type of data values a variable can hold and the operations that can be performed on that data. Data type is a category of values.

The built in data types are, Numbers (Integer type, Floating Point type, and Complex type). String, List, Tuple, Set, Boolean and None types

a) Integer Type

An integer type (int) represents signed whole numbers. An integer represents both positive and negative numbers. The range is at least -2,147,483,648 to 2,147,483,647. To write an int constant.

- A zero is written as just
- To write an integer in decimal (base 10), the first digit must be not zero.
  Example : 25000
- To write an integer in octal (base 8), precede it with "0o" and use the digits 0 to 7, plus A, B, C, D, E, F
  Example : 0o177
- To write an integer in hexadecimal (base 16), precedes it with "0x" or "0X".
  Example : 0x77

- To write an integer in binary (base 2), precede it with "0b" or "0B".
  Example : 0b101

b) Floating Point Type

A floating point (float) type represents numbers with fractional part. A floating point numbers has a decimal point and a fractional part.

Example : 3.0 or 3.17 or -28.72.

Python cannot represent very large or very small numbers, and the precision is limited to only about 14 digits. The floating point also represents scientific notation. It is stored with three parts.

- A sign + or -
- A mantissa
- An exponent

Example : 1.6E3 stands for 1.6×10^3, i.e., it is the same as 1600.0

c) Complex Type

The complex data type is an immutable type that holds a pair of floats, one representing the real part and the other representing the imaginary part of a complex number. Complex numbers are written with the real and imaginary parts joined by a + or - sign, and with the imaginary part followed by a j.

Example : 5 + 14 j

Python displays complex numbers in parentheses when they have a nonzero real part.

Example :

```
z = 5+14j
print(z.real, "and", z.imag)
print((2+3j) * (4+5j))
```

Output :

```
5.0 and 14.0
(-7+22j)
```

Program 1 : Illustration of number type

```
a = 0b1010  # Binary Literals
b = 100 # Decimal Literals
c = 0o310 # Octal Literals
d = 0x12c # Hexadecimal Literals
#Float Type
float_1 = 10.5
float_2 = 1.5e2
#Complex Type
x = 3.14j
print(a,b,c,d)
Print(float_1, float_2)
Print(x, x.imag, x.real)
```

Output :

```
10 100 200 300
10.5 150.0
3.14j 3.14 0.0
```

d) Boolean Type

A Boolean type represents special values 'True' and 'False'. They are represented as 1 and 0, and can be used in numeric expressions as value. The most common way to produce a Boolean value is with a relational operator.

Example : 2 < 3 is True

Program 2 : Illustration of Boolean type

```
print (1 == True)
Print (1 == False)
print (True + 4)
Print (False + 12)
```

Output :

```
True
False
5
12
```

In python, True represents the value as 1 and False as 0.

e) String Type

A string type represents sequence of characters surrounded by quotes. These characters may be alphabets, digits or special characters including spaces. In python strings values are enclosed in single quotes, double quotes or triple quotes.

When using triple quotes, strings can specify multi-line string without using escape character. The quotes are not part of the string; they are used to mark the beginning

and end of the string for the interpreter.

Example :

```
print ('Hello')
print ("Hello")
print ("""Hi,
        How are you ?""")
```

f) List Type

List is a sequence data type. List is a sequence of items separated by commas and the items are enclosed in square brackets []. The sequence of values can be of any type. The values in a list are called elements or items.

Example :

```
Address = ['231', 'Carmel Nagar', 'Nagercoil']
print (Address)
```

- The built-in python function "type" can be used to find out the type of an object.

Example :

```
print (type (2))
print (type (3.0))
print ("Hello")
print (type [1,2,3])
```

g) Tuples

A tuple is another sequence data type similar to list. A tuple consists of a sequence of items separated by commas and items are enclosed in parenthesis ().

The main differences between list and tuples are: Lists are enclosed in brackets [] and their elements and size can be changed, while tuples are enclosed in parenthesis () and cannot be changed.

Example :

```
T1 = ('Sun', 'mon', 'Tue')
T2 = ('P', 'Y', 'T', 'H', 'O', 'N')
print (T1)
Print (T2)
```

h) Sets

Set is an unordered collection of items seperated by commas and items are enclosed in curly braces {}. A set is similar to list, except that it cannot have duplicate entries (every element is unique). Once created, elements of a set cannot be changed.

Example :

```
x = set ("PYTHON PROGRAMMING")
print (x)
print (type (x))
```

Program 3 : Illustration of set type

```
a = {5, 2, 3, 1, 4}
print ("a = ", a)
print (type (a))
```

Output :

```
a = {1, 2, 3, 4, 5}
<class 'set'>
```

i) None Type

None is a speial data type with a sirgle value. Basically, the None data type means nonexistent, not known or empty. It is used to signify the absence of value in a situation. None supports no special operations, and it is neither False nor (0) Zero.

Program 4 : Illustration of None type

```
drink = "Available"
food = None
def menu(x):
    if x == drink :
        print(drink)
    else :
        print (food)
menu (drink)
menu (food)
```

Output :

```
Available
None
```