untitled0

September 24, 2024

1 Python Data Types

Python Data types are the classification or categorization of data items.It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in python programming,python data types are classes and variables are instances (objects) of these classes.The followimg are the standard or built-in data types in Python:

2 What is Python Data Types?

To define the values of various data types of Python and check their data types we use the type() function. Consider the following examples. This code assigns variable 'x' different values of various Python data types.It covers string, integer, float, complex, list, tuple, range, dictionary, set, frozenset, doolean, bytes, bytearray, memoryview, and the special value 'None'successively. Each assignment replaces the previous value, making 'x' take on the data type and value of the most recent assignment.

1.BASIC DATA TYPES:

***int**

This value is represented by int class. it contains positive or negative whole numbers (without fractions or decimals). In python, there is no limit to long an integer value can be.

***float**

This value is represented by the float class. It is a real number with a floating-point representation.It is specified by a decimal point.

COMPLEX NUMBERS:

Not only real numbers, Python can also handle complex numbers and its associated functions using the file "cmath". Complex numbers have their uses in many applications related to mathematics and python provides useful tools to handle and manipulate them. Converting real numbers to complex number An complex number is represented by " x + yi ". Python converts the real numbers x and y into complex using the function complex(x,y). The real part can be accessed using the function real() and imaginary part can be represented by imag().

BOOLEAN

Pyhton data type with one of the two built-in values, True or Flse. Boolean objects that are equal to true are truthy, and those eqyual to false are falsy. However non-boolean objects can be

evaluated in a boolean context as well and determined to be true or false. it is denoted by the class bool.

Note-

True and false with 'T' and 'F'are valid boolean otherwise python will throw an error.

Advanced data types:

Now that we've got the basics of strings and numbers down, let's talk about the advanced data types.

Iist Tuple *Dict* Set

EXAMPLE:

This code demonstrates how to determine the data type of variables in python using the type() function. it prints the data types of three variables : a (INTEGER), b(FLOAT), c(COMPLEX). The ouput shows the respective data type python for each variable.

[1]:
```
a = 10
print("\nType of a : ", type(a))

b = 23.5
print("\nType of b: ", type(b))

c = 3 + 7j
print("\nType of c: ", type(c))

e = -4 + 5j
print("\nType of e: ", type(e))
```

```
Type of a :  <class 'int'>

Type of b:  <class 'float'>

Type of c:  <class 'complex'>

Type of e:  <class 'complex'>
```

[4]:
```
#Boolean type
print(type(True))
print(type(False))
```

```
<class 'bool'>
<class 'bool'>
```

[5]:
```
#Example various of different types
a = 7
b = "Hello mama"
```

```
user_name = "karthik"
F = 2
E = 10
```

[6]:
```
#using type() to get data types
print(type(a))
print(type(b))
print(type(user_name))
print(F > E)
print(type(F > E))
```

```
<class 'int'>
<class 'str'>
<class 'str'>
False
<class 'bool'>
```

[8]:
```
E = 22
F = 56
print(E > F)
print(F < E)
print(E < F)
print(F == E)
print(E == F)
```

```
False
False
True
False
False
```

Conversion of one data type to another data type

[9]:
```
a = 4
b = 7.6
c = 10+6j
#convert int to float
x = float(a)

#convert int to int
y = int(b)

#convert int to complex
z = complex(a)

#convert float to camplex
f = complex(b)
```

```
print(x)
print(y)
print(z)
print(f)
```

```
4.0
7
(4+0j)
(7.6+0j)
```

Complex data types cannot be converted into any other data types

[10]:
```
#conversion of complex to int
a=2+4j
x=int(a)
printb(x)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-10-8ff029824fdb> in <cell line: 3>()
      1 #conversion of complex to int
      2 a=2+4j
----> 3 x=int(a)
      4 printb(x)

TypeError: int() argument must be a string, a bytes-like object or a real␣
 ↪number, not 'complex'
```