

8

Data Handling

In This Chapter

- 8.1 Introduction
- 8.2 Data Types
- 8.3 Mutable and Immutable Types
- 8.4 Operators

- 8.5 Expressions
- 8.6 Introduction to Python Standard Library Modules
- 8.7 Debugging

8.1 INTRODUCTION

In any language, there are some fundamentals you need to know before you can write even the most elementary programs. This chapter introduces some such fundamentals : *data types, variables, operators and expressions* in Python.

Python provides a predefined set of data types for handling the data it uses. Data can be stored in any of these data types. This chapter is going to discuss various types of data that you can store in Python. Of course, a program also needs a means to identify stored data.

So, this chapter shall also talk about mutable and immutable variables in Python.

8.2 DATA TYPES

Data can be of many types *e.g., character, integer, real, string* etc. Anything enclosed in quotes represents string data in Python. Numbers without fractions represent integer data. Numbers with fractions represent real data and *True* and *False* represent Boolean data. Since the data to be dealt with are of many types, a programming language must provide ways and facilities to handle all types of data.

Before you learn how you can process different types of data in Python, let us discuss various data-types supported in Python. In this discussion of data types, you'll be able to know Python's capabilities to handle a specific type of data, such as the memory space it allocates to hold a certain type of data and the range of values supported for a data type etc.

8.2.1 Numbers

As it is clear by the name the Number data types are used to store numeric values in Python. The Numbers in Python have following core data types :

(i) Integers

❖ Integers (signed)

❖ Booleans

(ii) Floating-Point Numbers

(iii) Complex Numbers

8.2.1A Integers

Integers are whole numbers such as 5, 39, 1917, 0 etc. They have no fractional parts. Integers are represented in Python by numeric values with no decimal point. Integers can be positive or negative, e.g., +12, -15, 3000 (missing + or - symbol means it is positive number).

Types of Integers in Python

- ❖ Integers (signed)
- ❖ Booleans

There are *two* types of integers in Python :

(i) **Integers (signed)**. It is the normal integer¹ representation of whole numbers. In Python 3.x can be of any length, it is only limited by the memory available. Unlike other languages, Python 3.x provides single data type (*int*) to store any integer, whether big or small. It is signed representation, i.e., the integers can be positive as well as negative.

(ii) **Booleans**. These represent the truth values *False* and *True*. The Boolean type is a subclass of integers, and Boolean values *False* and *True* behave like the values 0 and 1, respectively. To get the Boolean equivalent of 0 or 1, you can type *bool(0)* or *bool(1)*, Python will return *False* or *True* respectively. See figure below (left side).

```
>>> bool(0)
False

>>> bool(1)
True
```

```
>>> str(False)
'False'

>>> str(True)
'True'
```

NOTE

The `str()` function converts a value to string.

However, when you convert Boolean values *False* and *True* to a string, the strings 'False' or 'True' are returned, respectively. The `str()` function converts a value to string. See figure above (right side).

NOTE

The two objects `False` and `True` are the values for the `bool` type. `False` and `True` are Boolean objects.

8.2.1B Floating Point Numbers

A number having fractional part is a floating-point number. For example, 12.5 is a floating-point number. The decimal point signals that it is a floating-point number. The number 12 is an integer, but 12.0 is a floating-point number.