Modul 4

Exploring Python’s Data Types

1. Numbers

Number types: integers, shorts, longs, floats, and others. Python has expression operators that allow you to perform calculations on those numbers; these include +, - , / and *, %; comparison operators such as >, >=, and !=, or, and and; and many others.All of these operators are built in, but you can import others by using another of Python’s great characteristics: importing modules. Modules are extra libraries you can import into your script that add to Python’s native functionality. In this respect, Python is much like Java: if you want to do something, chances are very good that a library exists to make it easier. For example, if you want to parse text, such as web pages, you can check out the Beautiful Soup module. Need to log in to a remote computer (and you will, with some projects)? Import the telnetlib module and everything you need is available. And for numbers, the math module has all sorts of mathematical functions that add to Python’s number functionality. You can try it for yourself: in an IDLE session, type

```
MNM’“ 16:B2
```

and you should get the result 16. That’s because the absolute value function (I discuss the topic of functions in its own section later in this chapter) is already contained in Python’s default libraries. However, typing

```
MNM” 1:B7C2
```

will return an error, because the ceiling function is not in those default libraries. It must be imported. Now type

```
MNM ’
MNM ’ 7” 1:B7C2
```

and the terminal will return 17.0,—the ceiling of x, or the smallest integer greater than x. While you may not need to use the ceiling function, simply importing the math module gives you all kinds of extra functionality, such as logarithmic and trigonometric functions and angular conversions, all with just one line of code.

2. Strings

In Python, a string is defined as an ordered collection of characters used to represent text-based information. Python doesn't have a char type like C and other languages; a single character is simply a one-character string. Strings can contain anything viewable as text: letters, numbers, punctuation, program names, and so on. This means, of course, that

```
>>> x = 4
```

and

```
>>> x = "4"
```

are not the same thing. You can add 3 to the first example of x, but if you tried it with the second example, Python would give an error—that x points to a string with a value of 4, not an integer, because the 4 is enclosed in quotes.

Python does not distinguish between single and double quotes; you can enclose a string in either, and it will be recognized as a string. This has a nice side effect: you can enclose a quote character of the other type inside a string without having to escape with a backslash as you would have to in C. For example:

```
>>> "Brian's"
```

gives you

```
"Brian's"
```

without any escape characters needed.

There are some basic string operations you will probably use many times in your Python career, such as len (the length of a string), concatenation, iteration, indexing, and slicing (Python's equivalent of the substring operation). To illustrate, type the following bits of code into an IDLE session and see that the results match the output of what you see here:

```
>>> len('shrubbery') 9
'shrubbery' is 9 characters long.
>>> 'spam ' + 'and ' + 'eggs'
'spam and eggs'
```

'spam ', 'and ', and 'eggs' are concatenated

```
>>> title = "Meaning of Life"
>>> for c in title: print c,
```

The second through the fourth characters are 'pa'. (When naming a range of characters in a string, the first parameter is inclusive, the second is not.)

You can also convert to and from string objects for those times when you have an integer that is currently typed as a string, like “4”, and you need to square it. To do that, it’s as simple as typing

```
>>> int("4") ** 2
16
```

You can convert to and from ASCII code, format with escape characters like %d and %s, convert from uppercase to lowercase, and a whole host of other operations, just with Python’s built-in string library.

JOBSHEET 4

Lakukan penggunaan script program Python pada Raspberri Pi untuk menuliskan identitas anda di terminal Raspberry Pi.

LAPORAN HASIL PERCOBAAN: