Table Contents

Programming Refresher

Computer programming languages allow us to communicate with the computer and give instructions to a computer in a language that the computer understands. This document should serve as a refresher for the basic constructs of programming that will help learners to make better sense of things as they venture into the world of Python.

What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Most, programming is all about data: acquiring data, processing data, understanding data and this is where Python shines. The language was founded in the year 1991 by the developer Guido Van Rossum. It provides several packages and libraries for Data Analysis and Visualization, which makes it very flexible and fun to use. Python is widely used in bigger organizations because of its multiple programming paradigms and the syntax in Python helps the programmers to do coding in fewer steps as compared to other programming languages, and it's all free of cost.

Some useful link if you have no idea about Python so far:

https://www.python.org/doc/essays/blurb/

https://www.python.org/about/gettingstarted/

What is an Algorithm?

An algorithm is an executable sequence of elementary instructions, usually intended to accomplish a specific purpose and to perform a computation. A programming algorithm is a procedure that describes the exact steps needed for the computer to solve a problem or reach a goal. An algorithm includes calculations, reasoning, and data processing. Algorithms can be presented by natural languages, pseudocode, flowcharts, etc.

Here is a very quick read with some examples:

https://computer.howstuffworks.com/what-is-a-computer-algorithm.htm

What is a Variable?

Variables are used to store information to be referenced and manipulated in a computer program. They also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves. It is helpful to think of variables as containers

that hold information. Their sole purpose is to label and store data in memory. This data can then be used throughout your program.

For e.g.

Let a= 5 and b=10.

c = a+b =15

so here 'a', 'b' and are the variables, referring to the values or information 5, 10 and the addition of 5 and 10, while the addition is the operation. We can change the values from 5 and 10, but the operation will remain unaffected as 'a' and 'b' will associate themselves with the new values.

What are Data Types?

It specifies the type of values a data variable can take, and so drives the type of operations or calculations that can be applied to it. Following are the most common type of data variables:

Integers (int): As the name suggests, it can contain integers only and no fractions or decimals or non-numeric. Example: 1, 2, 345, 678

Float: It is to store fractions, decimals or scientific notations; character e or E followed by a positive or negative integer represents scientific notation. Example: 27.566, 2.8, .7e2

String(str): It is used to store characters or texts rather than numbers. It can also take numbers but they won't act as numbers i.e. you can't perform addition on a number stored in a string. Example: "Machine Learning"

Boolean (bool): This data type can take only two values - true or false. It can be represented by 1 or 0 as well.

What are Data Structures?

If you want to analyze data and process it, you have to store it somewhere and data structures are the solution. So a data structure is a particular format for organizing, processing, retrieving and storing data.

List: A list is a collection of other objects –floats, integers, complex numbers, strings or even other lists. A list is created by placing all the items (elements) inside a square bracket [ ], separated by commas.

Example: list = [1, "list", 22, 2.4]

https://www.w3schools.com/python/python_lists.asp

Dictionary: In python, a dictionary consists of key-value pairs. The value can be accessed by a unique key in the dictionary.

Example: dictionary = {" MARKS ": 29, "RANK" : 1}

https://www.w3schools.com/python/python_dictionaries.asp

Tuple: Tuples are similar to lists but they are immutable i.e. the elements comprising a tuple cannot be changed. It is not possible to add or remove elements from a tuple. A tuple is created by placing all the elements inside round brackets ().

Example: (1,20,36,4,5)

https://www.w3schools.com/python/python_tuples.asp

Array: An array is a data structure that stores a sequential collection of the same type of elements so an array maintains homogeneity. An array can be 1 dimensional or multidimensional. They are always rectangular so that all rows have the same number of elements.

https://docs.scipy.org/doc/numpy/reference/arrays.html

```
# importing the library numpy for arrays

import numpy as np

# creating an array

arr = np.array([[1],[3],[5]])

arr[2]  # displaying the third element of array
```

Matrix: A matrix is like a subset of array but it always has 2 dimensions and follows the rules of linear algebra.

```
# importing the library numpy for arrays

import numpy as np

# creating a matrix,which basically has rows and columns

matrix = np.array([[1,2],[3,4],[5,6]])  # The matrix has 3 rows and 2 columns.

matrix[2,1]    # This would give the element in the third row and second column.
```

https://www.w3schools.in/python-data-science/matrices-in-python/

Vectors:

By definition, Vectors are geometrical objects that have both magnitude and direction. It is defined by a line having an arrow the length of the line is its magnitude and the arrow specifies the direction. In python, vectors are one-dimensional dynamic arrays so unlike an array, it provides the ability to quickly change storage capacity.

```
# importing the library numpy for arrays

import numpy as np

# creating row vector

row_vector = np.array([1,2,3,4,5,6])

row_vector  # This would give the array in a row format.

# creating column vector

col_vector = np.array([1],[2],[3],[4])

col_vector    # This would give the array in a column format.
```

What is a Function?

A function is a set of code that is defined to perform a specific task. It is useful when one complex or large operation is to be done on different or the same variables again and again at different stages. We can call that function whenever we need it instead of writing the whole code again, so it reduces the effort by omitting the need to write the same code multiple times.

Argument: Arguments are the elements that we feed into a function and that function does the operation on those variables and gives us the required output.

e.g. we can define a function, say, *funny* = F (a,b) as A + B * A / B, then just writing *funny*(2,3) will give 2 + 3 * 2 / 3 = 4

What are Loops and control structures?

Loops: These are constructs that enable you to run the same piece of code many times, with different inputs and outputs, until a particular condition or criteria is met. These are iterative statements. It runs the code for each element in a sequence beginning with the 0th element and continuing until the final item. Two very common types of loops are - For loop and while loop.

Example: (for loop):

```
for i in [1,2,3]:

    print(i)
```

Here, it will assign print the value of i each time until all the values in that array are taken up by i. So, it will print 1, then 2, and then 3.

Example: (while loop):

```
i=1  # initializing the variable
while i < 5:   # while loop
    print(i)
    i +=1
```

Here, it will print i as long as i is less than 5. So, it will print 1, then 2, then 3 and then 4.

if-else statements: These are conditional statements that enable us to check the condition and enable us to change the behavior of the program which changes the output accordingly.

```
a = 1
if a ==1:
    print(a)
else:
    print(" wrong value")
```

so if the value of a is 1 then we will get the output as 1 but if the value of is not 1 we will get the output "wrong value". https://www.w3schools.com/python/python_conditions.asp

What is a Syntax?

There are certain rules to write codes and we have to follow those rules for the successful execution of the codes. For example, while writing the if-else statement in Python a fixed syntax is followed so that the machine can understand the code. Syntax is like the grammar of any coding language and if it is not correct it will change the context of the code and the compiler will fail to execute it.

Are there some good resources to learn from?

Yes, there are plenty of resources available for free on the internet. Here are a few of them.

1. A resource that mentions other resources
2. A very good resource for beginners in python for data science
3. A gentle introduction to python for those with programming background
4. If you don't want to stop just at the beginner's stage
5. This is for hardcore coders
6. Another one with everything (almost)