STUDY REPORT

Lov Dubey

September 2022

1 Introduction To Overleaf

Overleaf provides a rich-text editor so you don't need to know any code to get started—you can just edit the text, add images, and see the typeset document automatically created for you as you type. Our tutorial provides a quick three-step introduction to the main features.

2 Introduction To Python

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

3 Introduction To Google Colab

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs

4 Famous functions of python for data science

A function is a collection of related assertions that performs a mathematical, analytical, or evaluative operation. Python functions are simple to define and

essential to intermediate-level programming. The exact criteria hold to function names as they do to variable names. The goal is to group up certain often performed actions and define a function. Rather than rewriting the same code block over and over for varied input variables, we may call the function and repurpose the code included within it with different variables.

The functions are broad of two types, user-defined and built-in functions. It aids in keeping the software succinct, non-repetitive, and well-organized.

4.1 print( ) function

The print() function prints the specified message to the screen or another standard output device.

The message that wants to print can be a string or any other object. This function converts the object into a string before written to the screen.

4.2 input( ) function

The input() function allows taking the input from the user.

4.3 abs( ) function

The abs() function returns the absolute value of the specified number.

4.4 pow( ) function

The pow() function returns the calculated value of x to the power of y i.e, xy.

If a third parameter is present in this function, then it returns x to the power of y, modulus z.

4.5 dir( ) function

The dir() function returns all the properties and methods of the specified object, without the values.

This function even returns built-in properties which are the default for all objects.

4.6 sorted( ) function

The sorted() function returns a sorted list of the specified iterable object.

You can specify the order to be either ascending or descending. In this function, Strings are sorted alphabetically, and numbers are sorted numerically.

NOTE: If a list contains BOTH string values AND numeric values, then we cannot sort it.

4.7 max( ) function

The max() function returns the item with the maximum value or the item with the maximum value in an iterable.

If the values this function takes are strings, then it is done using an alphabetical comparison.

4.8 round( ) function

The round() function returns a floating-point number that is a rounded version of the specified number, with the specified number of decimals.

The default number of decimals is 0, meaning that the function will return the nearest integer.

4.9 divmod( ) function

The divmod() function returns a tuple containing the quotient and the remainder when the first argument i.e, the dividend is divided by the second argument i.e, the divisor.

4.10 id( ) function

The id() function returns a unique id for the specified object. Note that all the objects in Python have their own unique id.

The id is assigned to the object when it is created.

The id is the object's memory address and will be different for each time you run the program. (except for some object that has a constant unique id, like integers from -5 to 256)

4.11 ord( ) function

The ord() function returns the number representing the Unicode code of a specified character.

4.12 len( ) function

The len() function returns the count of items present in a specified object.

When the object is a string, then the len() function returns the number of characters present in that string.

4.13 sum( ) function

The sum() function returns a number, the sum of all items in an iterable.

If you are a beginner, I think that you don't have a good understanding of what Iterables and Iterators are. To learn these concepts, you can refer to the

4.14 help( ) function

The help() function is used to display the documentation of modules, functions, classes, keywords, etc.

If we don't give an argument to the help function, then the interactive help utility starts up on the console.

5 Features of Python

5.1 Easy to Learn

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

5.2 Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

5.3 Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

5.4 Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5.5 Free and Open Source

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

5.6 Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

5.7 Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

5.8 Large Standard Library

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

5.9 GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

5.10 Integrated

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C,C++ Java. It makes easy to debug the code.

5.11 Embeddable

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

5.12 Dynamic Memory Allocation

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 15 to x, then we don't need to write int x = 15. Just write x = 15.