Chapter 1: Basics of Python

Objectives

To discuss the basic of python by exploring the

- Meaning and application of python
- Discussing the outlook of lanaguage by mentioning about its interpreter and code editor
- Explaining the basic terminologies and elements of Python by mentioning about procedure of working with python and the keywords used.

Introduction

Python is simply an interactive, general-purpose interpreted, high-level, and object-oriented programming language which was released in 1991 and created by Guido van Rossum. Rossum wanted a mysterious and unique name for the programming language and thus, named it Pythonafter the BBC comedy series of Monty's Python's Flying Circus. With the basic version released in 1991, the updates of language started in January 1994, and the latest version named Python 3.10.11 was released on 5th April 2023. Being a continuously evolving and easy-to-learn language, python is consistently recognized as the most popular language in the world. This chapter will focus on digging deep into the concept of Python by assessing the application of Python, its interpreter, code editors, and writing in the introductory program. Further, basic terminologies and working in Python would be discussed in detail to provide an overview of how a simple program could be returned and what all different elements are included while writing the program.

Application of python

In today's time, we are living in a world wherein the usage of software is the major requirement for all types of industries like the military, research, banking, or healthcare. There is a huge list of programming languages available for supporting the requirement but among them, Python is the most exciting and lucrative programming language. Python is a language that is simple to understand and use due to its syntax similar to the English language, free and open source, has the presence of extensive libraries for supporting every imaginable function, is interpreted language which could be read line by line and could be dynamically typed. These features made Python among the most popular programming language and resulted in having its demand by all top companies. In the real world, python could be used for supporting software components, and website development, or to work with machine learning (ML), artificial intelligence (AI), and data science technologies. The top applications of Python in the real world are discussed below.

Desktop GUI

GUI (Graphical user interface) is the interface that helps in building user interaction with the electronics using audio indicators or graphical icons instead of having text-based details like text navigation of typed command use. As Python is an interactive language thus it helps in building the GUI for the user easily and quickly. There is the presence of many inbuilt tools in Python

such as wxWidgets, kivy, or PyQT library which helps in developing functional GUI efficiently and securely.

Console Based

The console-based application could be stated as the command program which can be used for the execution of the program. In old computer generations, console-based applications are very famous. Python has Read Eval Print Loop (REPL) which makes the language suitable for having command-line applications. There is the existence of many IO libraries with Python which contribute towards building command line apps.

Software development

The presence of features like high compatibility, enhanced code readability and reusability, the presence of many inbuilt frameworks or libraries, and platform independence contributes towards making Python suitable for software development. Even there is the existence of technologies like AI and ML which can be integrated with Python for software development. Some of the common applications that are using Python are Reddit, Google, and Netflix.

Scientific and numeric

Python is identified as the means of simplifying all scientific computing by the presence of libraries like NumPy (for addressing complex numeric calculations), IPython (for recording and editing work sessions and supporting parallel computing and visualizations), Pandas (for data modeling and analysis), and SciPy (package for supporting engineering, science and mathematics work).

Business

Python helps in providing scalability and security features which helps in the development of high-performing business applications. There is the existence of libraries like Tryton which is an open-source business software with functions like purchasing, financial accounting, shipping, or sales thus making the language suitable for business applications.

Audio or video-based

The applications for audio and video are the most interesting feature of Python wherein there is the presence of libraries like Pyo, SciPy, Mingus, Dejavu, and OpenCV which helps in completing the task flawlessly. Some of the applications which are formulated using Python are YouTube, Spotify, and Netflix.

CAD

CAD (Computer aided design) is defined as the procedure which is used for the creation of 3D or 2D models digitally. The CAD applications majorly are used by construction managers, product designers, or architects for maintaining consistency and designing products. Python has tools like Open Cascade, FreeCAD, or Blender which provide features of dynamic system development,

technical drawing, file exporting or importing, and recordings. These efforts help in designing the products efficiently.

Enterprise

Within an organization or an enterprise too, the Python language could be to support operating needs. Herein, the real-time usage of Python for enterprise applications is like Picalo, Tryton, or OpenERP.

Image processing

Python is identified as an important tool for image analysis and manipulation. With the presence of open-source dependencies that can be installed using the command line, python helps in the assessment of the images using tools like Pillow, OpenCV, or SimpleITK.

Web

The web applications for Python focus on the development of webpages using Python tools. There is the existence of many protocols with Python like JSON, XML, HTML, BeautifulSoup, Request, or email processing which helps in web development. There is the presence of frameworks like Bootle, Flask, or Django which are used by developers for making the process completely effortless. Further, the usage of Python also contributes to having a fast development process, enhancement in security, adding convenience in development, and supporting better visualization.

Python in the real world has bought many changes to industries. As there is the presence of many powerful libraries which helps in fulfilling the development requirements of industries, thus, usage of Python is essential for boosting productivity.

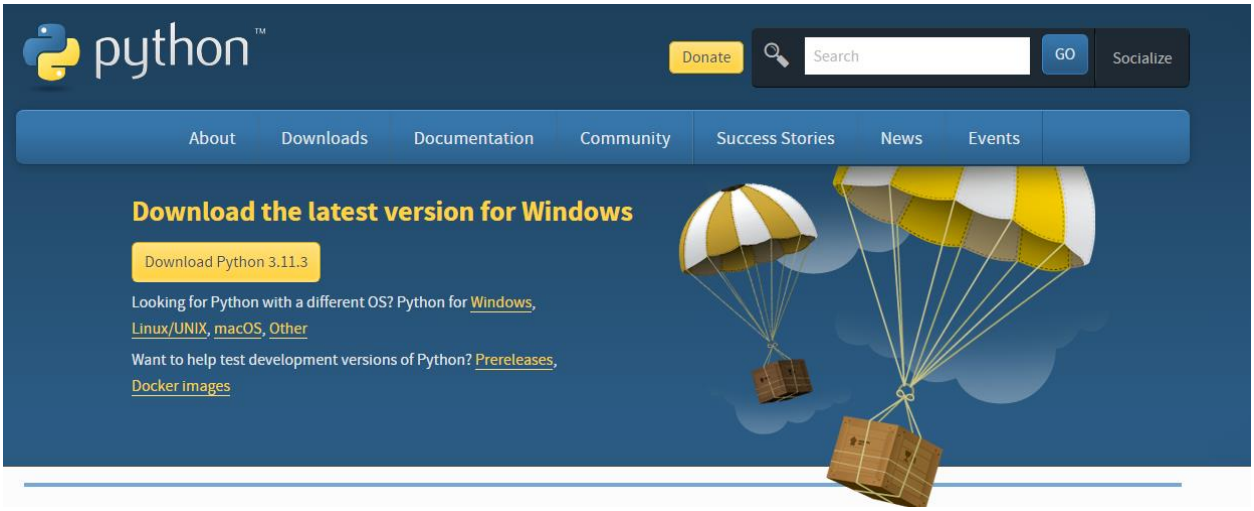
Using Python Interpreter

Python interpreter is defined as a computer program that helps in converting a high-end program into machine language. The interpreter is the tool used for translating the program that is typed into the language which the computer could understand. As machine language is represented by bits strings i.e. 1s or 0s and reading this language is difficult for humans. Thus, to bridge the gap and make the codes readable for humans and understandable for computers, interpreters are used. The interpreter is different from the compilers. Though the compiler also contributes to translating but the functioning is different. With an interpreter line to line, translation is done while with a compiler batch translation is practiced. This results in having less time requirement with the interpreter. Usually, the Python interpreter is stated as

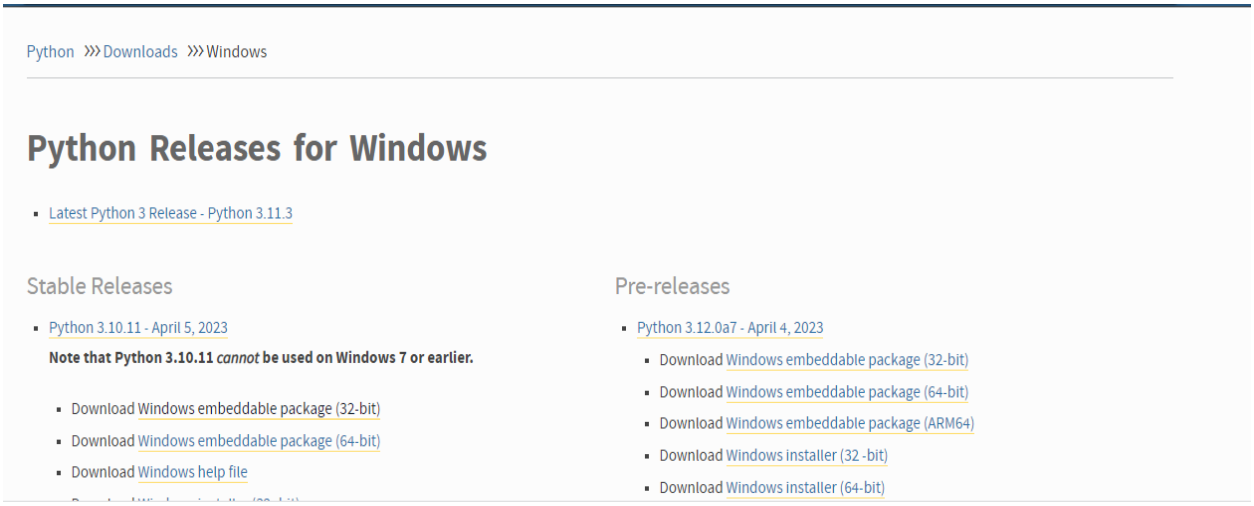
/usr/local/bin/python3.11

As the choice of the directory where the interpreter is installed is optional, thus, other places could also be used for installation but mostly the installation is present on the stated path only.

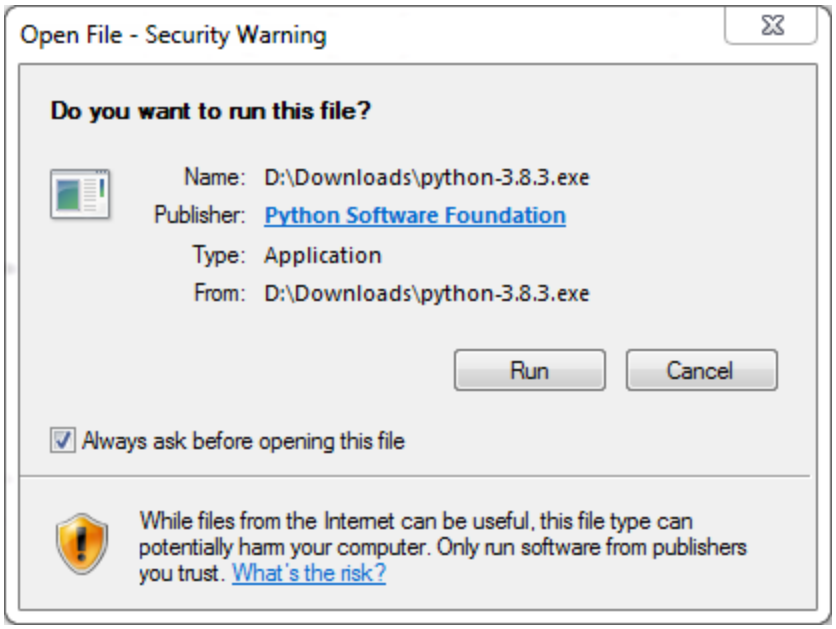
For installing in Python, visit https://www.python.org/downloads/ and download the latest version of Python for your system. Currently, the latest version available for Windows is 3.11.3. Now select the system type i.e. Windows, macOS, Linus/UNIX, or any other



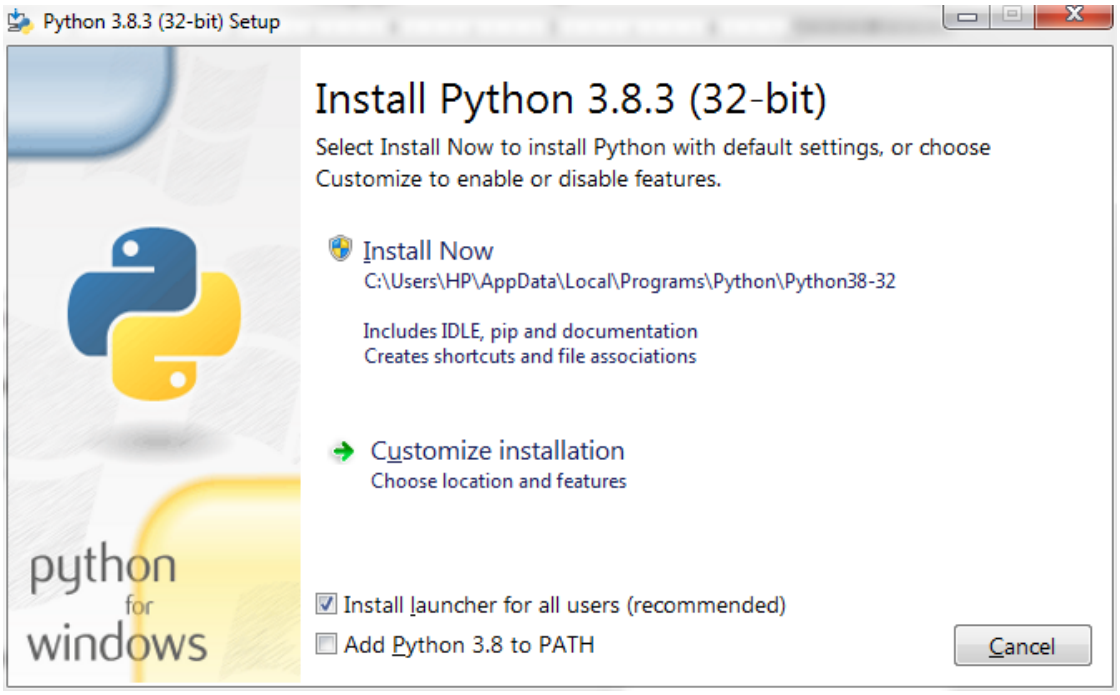
Select the Python version you want to download,



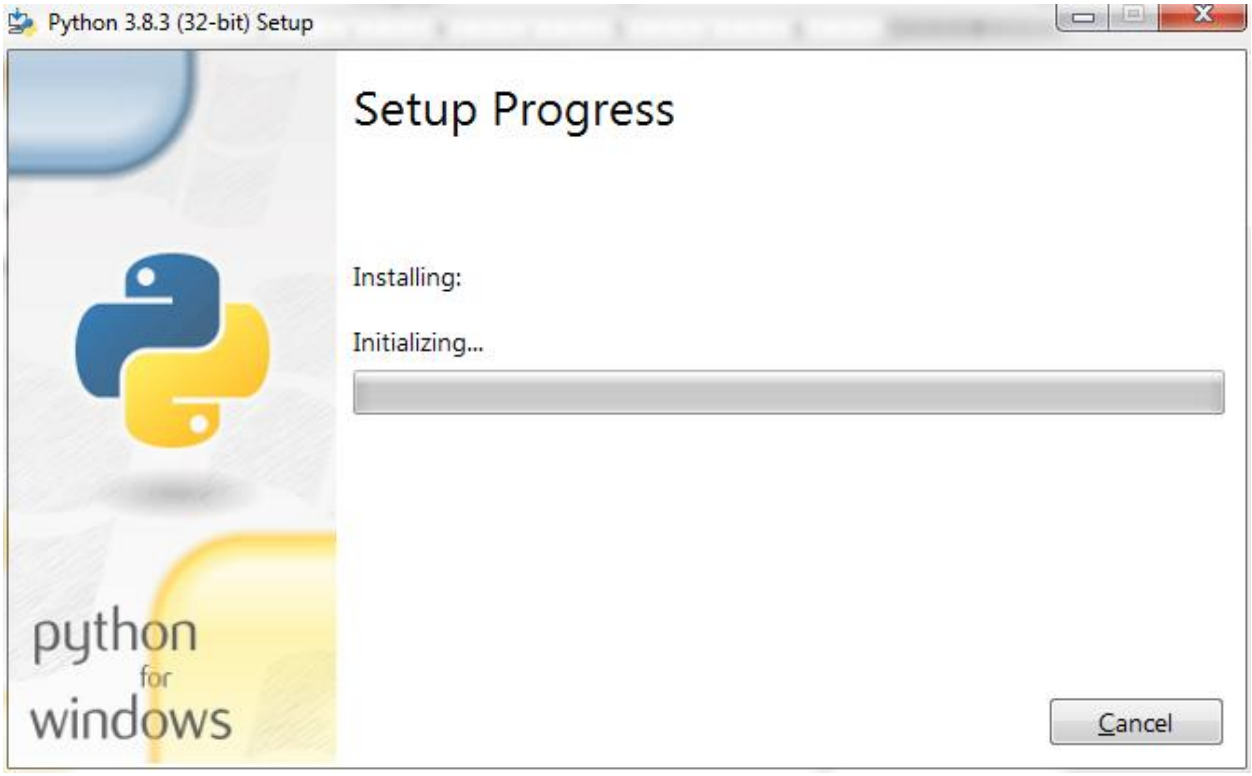
click on the exe file and download it. Suppose the 3.8.3 version is downloaded then



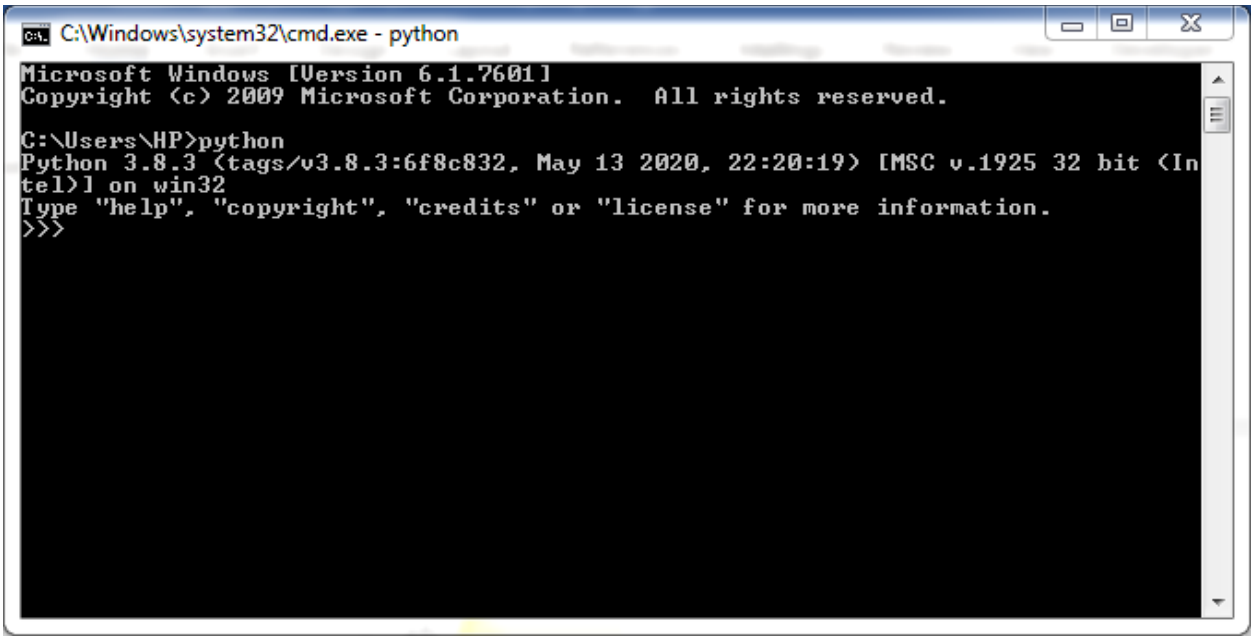
Once the file is downloaded open the file and run it for installation.



By clicking on Install Now, the Python interface would be installed for Windows. With this below window will appear



With this, a Python interpreter would be installed which you can check by opening the command prompt and typing python.



The Python command represents the version that is present in the system and some of the built-in functions which can be used for more information about the downloaded version of Python.

There are two methods for using in the Python interpreter i.e.

- Argument passing – The method focuses on giving in the argument based on which the execution is required. Herein, the additional arguments and string names are converted into lists. For example, by importing the sys module, the length of the module is at least 1 with no arguments and strings presence. Now if the -m module is used then the sys.argv[0] is set to be as the located module full name.
- Interactive mode – The interactive mode involves command reading using tty. Herein, the next command is based on the primary prompt thus, >>> signs are used.

Code editors in Python

A Python code editor is defined as an editor which is used specifically for writing or editing a program. As often in the real world, the program written is very lengthy and complex which makes the working with Python command line difficult. Thus, for keeping track of the entire code and having easy modification and writing, code editors are used. There are many code editors which are popularly used i.e. IDLE, Sublime Text, PyCharm, Visual studio code, Atom, Jupyter, Spyder, PyDev, thonny, or Wing. Among the stated code editors, IDLE is among the most popular code editor as it is suitable for beginners, is free, and is a very capable debugger. Simplerly PyCharm, Sublime Tex, and Visual Studio codes are also free and have features of good functioning, thus, they are also among the popular editors used for Python.

Introductory program of Python

Let's start working with Python by writing a very simple program of “Hello World!”. Herein Sublime Text is used as the script for the program. So, as the purpose is to just print the statement, thus, the code is designed by using the print function i.e.

```
print("Hello World!")
```

The code herein is written on the code editor thus, for running the command and deriving the result, the python command line will be used. So, opening in the command prompt and selecting the directory wherein the text file is saved using the “cd” command and using backslash as the means of defining in sub-folder, the python file is executed. For this, type the file name followed by the file type and click on enter i.e.

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
Hello World!
```

The output in the Python command line is derived of Hello World!.

Getting Started with Python

For many computers python is already installed, thus, we can check if Python is installed on Windows PC or not by searching on the start bar for Python or running it on the command line

In case Python is not installed, then using the versions published on the official website of Python https://www.python.org/, the language could be downloaded.

Python script

Python is the interpreted programming language focused on having the code written in Python i.e. .py files in text editors. These files are then placed into a Python interpreter for execution. For running in a Python file developed in a text editor and named sample.py i.e.

```
print("Hello everyone!")
```

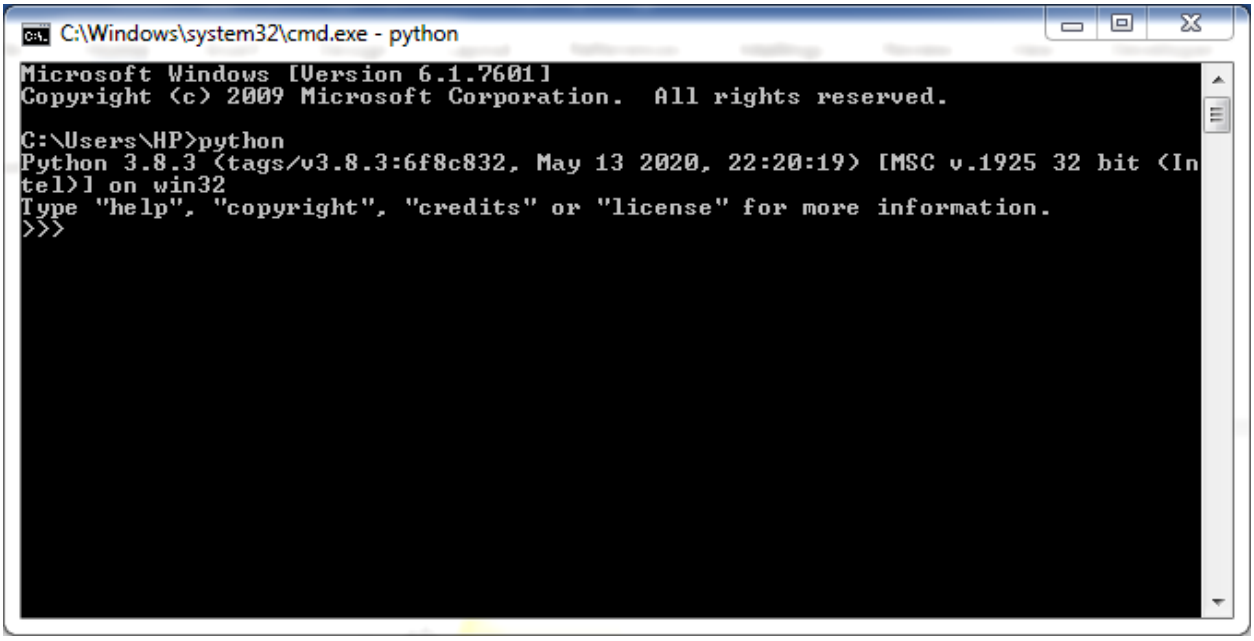
we give the command in the command prompt

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
Hello everyone!
```

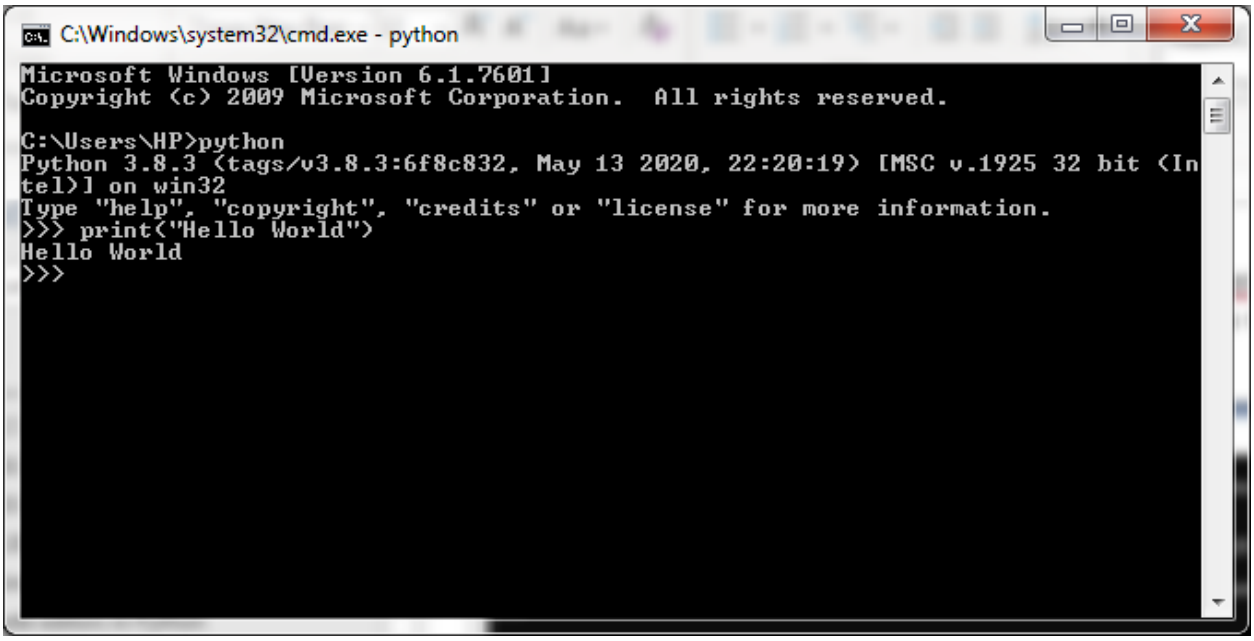
The file which contains all the commands which need to be processed and are structured in the form of a program is known as Python script. Thus, the Python file saved on the text editor is simply called Python script. As the procedure of developing complex programs with lengthy steps is time-consuming and difficult, thus, to simplify the situation, python scripts are essential.

Python command line

For testing in the shortcode sometimes instead of running a Python script, the code could be directly written on the Python command line. This could be as with command prompt, python could itself be run as a command line like



Herein, we can write a simple introductory program like printing hello world could be used.



For exiting in this command line the exit() function can be used.

Print function

The print function defined by print() is used in Python for printing the specified message. This message could be any object, string, or any other thing written in the code. The syntax for the print function is

$$print(object(s), sep = separator, end = end, file = file, flush = flush)$$

Wherein the object defines anything which needs to be printed, the separator is the statement of how to separate the objects, the end mentions the point wherein the print will end, the file defines the object with the write method, and flush is a boolean used for mentioning that whether the stated object is True or False.

```
print("I am","", "learning Python")
```

The output for the command would be

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
I am  learning Python
```

Herein, space is added as the separator.

Python syntax

For every programming language, there is a set of rules defined for writing the program. These rules are known as syntax. As initially the code written with Python is read by a parser and the parser would only be able to understand the code if it's written as per the rules, thus, it is essential to understand language syntax. Python syntax consists of

- Line structure wherein logical lines are added and all blank lines are ignored. New lines mean new statements with python
- The multiline statement defines the method wherein more than one line is used for single statement representation. Herein backward slash or triple quotes could be used

```
print("Hello\
            World!")
print("""I am
            good""")
```

The output for the above code is

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
Hello           World!
I am
                good
```

- Multiple statements in one line could be stated by separating the statement using semicolons

```
a=1;print(a)
```

The output for the above statement is

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
1
```

- For writing string, the single or double quote could be used but just keep the format the same i.e. if started with a single string then end also with a single string.

```
a = 'Finance'
print(a)
```

The output is

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
Finance
```

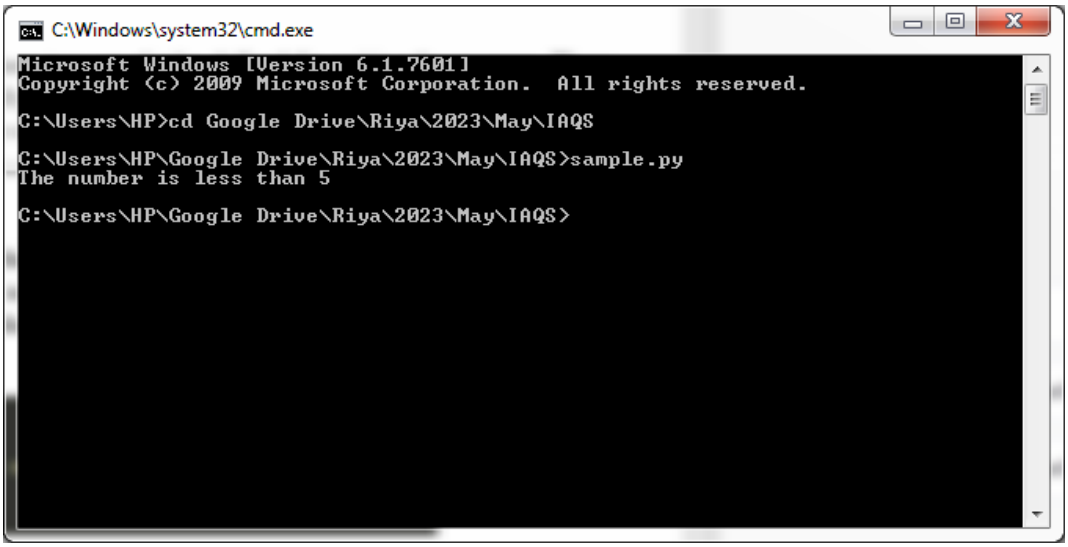
- All the blank lines or whitespaces are ignored by the interpreter

Python indentation

Indentation is defined as the space which is added at the beginning of the code. Though for other programming languages indentation is just a means of presentation but with Python, indentation is very important as it defines the block for a function or method i.e. indentation in Python is used for block creation.

```
a = 0
if a<5:
    print("The number is less than 5")
    a = a+1
```

With this indentation, the result would be



In case indentation is not properly placed like in the if statement no indent is added, then a syntax error is derived.

```
a = 0
if a<5:
print("The number is less than 5")
a = a+1
```

The output would be

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
  File "C:\Users\HP\Google Drive\Riya\2023\May\IAQS\sample.py", line 3
    print("The number is less than 5")
    ^
IndentationError: expected an indented block
```

Herein, an indentation error is derived representing that the syntax of the code is not properly indented.

Though there is no restriction on the number of spaces used for indentation but the thing to be considered is that the number of spaces should remain consistent throughout the block.

Creating comments in Python

The python has capability for stating in comments in the code. Comments are simply statements used for explaining the code and are not included as part of the code. The comments help in making the entire code more readable and prevent time wastage in understanding the code while executing or re-checking in case of error or after a long time. The comments in Python are created using the"#" command and Python will ignore the statement written after it. For multiple line strings the double quotes i.e. "" could also be used to add comments.

```
a = 0
#the loop is created for running the command until the value is less than 5
while a < 5:
    print("The number is okay")
    a = a + 1
```

Herein, the comment is stated to explain what would be done in the while loop.

The output for the

```
C:\Users\HP\Google Drive\Riya\2023\May\IAQS>sample.py
The number is okay
The number is okay
The number is okay
The number is okay
The number is okay
```

As 5 times, the value was less than 5 thus, the statement ”The number is okay” is printed 5 times.

Basic terminologies with Python

Though the code varies in Python as per the requirement but some of the terminologies are common which are required for each program building. Some of the basic terms used with Python are

- Variables – Python variables are simply the containers used for storing the data values. There is no command present in Python for declaring a variable. The variable is developed with the assignment of values to it.

```
name = "Ranjan"
age = 20
print(name,age)
```

  In the above example, the name is the variable created to store a string value Ranjan while age is the variable developed for storing integer value i.e. 20. Thus, there is no requirement of declaring a variable for any particular type only, we can change them once a value is set.

- Function – A function in Python is defined as a block of code that only can run when it's called. It is a re-useful form of code that can be used again for performing a particular action. Instead of copy-pasting the same line of code again and again, the complex code could be broken down into more maintainable and readable forms. Herein the data (which is known as parameters) could be parsed into a function and the data could be derived from the function as a result.

```
def feature():
    print("I am learning Python")
feature()
```

- Classes – Python classes are defined as the prototype or blueprint wherein the objects are created. A class simply is the means of having functionality and data bundling. It is useful for creating a user-defined data structure that includes its own data members and functions. The class is developed using keyword class and herein variables developed inside the class are the attributes. For example, a class need to be developed for student

wherein multiple elements are present like course, age, semester, and department. Now, for recording the BBA course we can develop a class student with a course as BBA i.e.

```
class student:
    course = "BBA"
```

- Object – Python object is an instance of the class. Herein class is like an entire blueprint while the object is a class copy but with the inclusion of actual values. It's not any idea but now the details are specific. The object would include a state (object attribute or the object properties), behavior (object methods or one object response to another), and identity (the unique name for any object used for interacting with other objects).

```
class student:
    course = "BBA"
    age = "20"
    def value (self):
        print("My course is",self.course)
        print("I am", self.age)

Neha = student()
Neha.value()
```

For the given code, the output would be

```
My course is BBA
I am 20
```

Herein a class is created named student having 2 attributes known as class variables i.e. course and age. A method named value is developed which helps in printing the values. Now an object named Neha is created from the student class and we are calling the value method to print these values.

- Lamba – Python lambda is the keyword that helps create the lambda functions. These are the anonymous functions that can be used for simple calculations. For example – With the inclusion of the x/y function, a lambda function is developed

```
a = lambda x, y : x/y
print(a(6,2))
```

The output for the code is

```
3.0
```

Herein, the value of a variable is derived to be 3.

- Array – Python array is defined as the specific variable which is used for storing more than one value at a time. As more than 1 value is recorded thus, their index values could be used for referring to the number. For example – There is a list of insurance provided by an insurance company i.e. life insurance, health insurance, general insurance, home insurance, and property insurance. Then the code would be

```
insurance = ["home","health","life","general","property"]
print(insurance[1])
```

Herein, insurance is the array with all types of insurance mentioned. For assessing the insurance at an index value of 1, we print the value and the output is health.

```
health
```

Python Keywords

Similar to any other programming language, python also includes some specified words which have restrictions on where those words could be used. These keywords serve as the building block for the program and thus a programmer needs to be aware of the keywords for creating a program.

What are keywords

The keywords are defined as the reserve words which have some specific meaning and use in the programming language and these words could not be used for any other thing. Python has the presence of these keywords and no requirement is there of importing them. Though built-in function in Python is also present but as there is no such restriction exists for them as for keywords, thus, they are different from keywords. For Python 3.8, there is the existence of 35 keywords with some of them being False, and, or, None, finally, yield, else, as, pass, or in.

How to identify Python Keywords

The Python keywords can be identified using three means i.e. IDE, code in REPL, and syntax error.

- With IDE, all the keywords are highlighted and it would help to differentiate them from other words.
- Using REPL, the keywords could be used by help(). This would provide a list of all keywords. However, for specific keyword information, the word could be passed through a help function like help(”from”). Even a keyword module exists from which the details about keywords could

be derived i.e. kwlist and iskeyword() for knowing whether a particular work is a keyword or not.

- Lastly, the syntax error derivation when a value is assigned to the keywords or the word is used as a function name also depicts that the keyword is used incorrectly.

Most common Python Keywords (Value, Operator, Iteration, Nonlocal, Exception handling, pass, return, and many more)

There are multiple keywords present in Python and based on the nature and usage of them, the keywords could be grouped

- Value keywords – The value keywords are the ones consisting of singleton values and can be used multiple times. The keywords are True, False, and None.
    - *True* Keyword is the Boolean true value
    - *False* is the Boolean false value.
    - *None* Keyword on the other hand represents the existence of no value and is the default value if no return statement is present.
- Operator keywords – Many operators in Python could be used as keywords like AND or ∧ as and, Not or ¬ as not, OR or ∨ as or, IDENTITY as is, and CONTAINS or ∈ as in.
    - *and* keywords herein are used to know whether both right and left statements are falsy or truthy. If both are true then they are truthy else a falsy
    - *or* keyword is used to check the truthiness of at least one of the statements i.e. if one of the statements (right or left) is truthy then the result is truthy.
    - *not* keyword is used for defining opposite boolean value i.e. conditional statement used for flipping the meaning.
    - *in* keyword is membership operator or containment check keyword used to know whether a particular element is present in the specific variable or set or not.
    - *is* keyword a means of having an identity check i.e. to know whether two objects are exactly the same or not
- Control flow keywords – The keywords used for controlling the flow are called control flow keywords.
    - *if* keyword is the conditional statement used for building a code block and which operates only when the stated condition is truthy.
    - *else* keyword is the Python keyword *elif* or *if* conjunction which also denotes code block and is used only when the elif or if condition block is not truthy.
    - *elif* keyword is used for adding multiple conditions i.e. can be used only after an if or elif statement and there are no restrictions on the number of *elif* usage.
- Iteration keywords

- *for* keyword is combined with *in* keyword and used for specifying the condition till when the loop should continue.
- *while* keyword is used to continue the iteration of the loop until the while keyword condition is falsy.
- *break* keyword helps in exiting the loop earlier than the normal existing
- *continue* keyword is used for skipping in the iteration of the next loop
- *else* keyword with loops is used to define that loop will continue to run when the loop normally exists i.e. when the break is not called.

- Structure keywords – For defining classes or functions in context manager, the structure keywords could be used.
  - *def* keyword is used for defining a method or function of a class.
  - *class* keyword is the one used for defining in the class
  - *with* keyword is used for defining in the code which needs to be executed under the scope of context manager
  - *as* a keyword with *with* is used for accessing the values derived using with the keyword by using as alias.
  - *pass* keyword is used for stating in that a block is intentionally kept blank. Its like no operation.
  - *lambda* keyword is used for defining the function which has no name but just one statement and based on it the results are derived. A function formulated with the *lambda* keyword is called a lambda function.

- Returning keywords – The keywords used for defining the values which should be returned from the methods or function are known as returning keywords.
  - *return* keyword is a keyword defined with def and which is used for exiting the function and returning the value or result specified after the return keyword.
  - *yield* keyword is similar to the return keyword which mentions what needs to be returned from the function. However, the value derived from the yield keyword is a generator and could be parsed with built-in-next() function for having the next value generation from the function.

- Import keywords – There are many libraries or modules which are not in-built and hence need to be imported into the program. The keywords used for importing are
  - *import* keyword is a keyword used for importing or including a module from Python
  - *from* keyword is used for having import something specific from a particular module

    - *as* keyword is an alias used for importing a tool or module. The keyword is used along with from and import keywords. For lengthy and complex names, the keyword help in naming the keyword as something else which is simpler.

- Exception handling keywords – The keywords used for catching and raising exceptions is exception handling keywords. Some of these keywords are
    - *try* keyword used for raising exceptions and defining what is done if an exception is raised.
    - *except* keyword is used along with *try* to define what can be done if some specific exemptions are raised. With a single try, more than one except blocks could be created.
    - *raise* keyword is used for exception raising
    - *finally* keyword is the specification of code that needs to be compulsorily performed irrespective of the results in *else, except,* and *try* block.
    - *else* keyword used along with *try* and *except* means than else keyword could only be used if an exception is not raised with a *try* block.
    - *assert* keyword is used for specifying in the assert statement or having any expression assertion. Herein no-op would be derived if the expression is truthy while an assertion error is drawn when the expression is falsy.
- Variable handling keywords -The keywords used for working with variables are known as variable handling keywords. Some of the common keywords are del, global, and nonlocal.
    - *del* keyword is used for unsetting a name or variable and commonly the keyword is used for list or dictionary index removal.
    - *global* keyword is the keyword used for defining a variable in global scope i.e. a variable that can be pulled at any level in the code.
    - *nonlocal* keyword is like a *global* keyword used for variables modification from the global scope. Herein nonlocal keyword helps in pulling the variable from the parent scope.

Chapter Summary

Python is simply an interactive, general-purpose interpreted, high-level, and object-oriented programming language

In the real world, python could be used for web applications, image processing, enterprise application, CAD, audio or video-based applications, business applications, scientific and numeric, software development, console-based and desktop GUI

Many code editors are present for Python, but among them, IDLE, PyCharm, Sublime Tex, and Visual Studio codes are the most popular

For writing in the program in Python, the requirement is to be aware of some basic terms like class, variable, object, array, and lambda

The keywords are defined as the reserve words which have some specific meaning and use in the programming language and these words could not be used for any other thing. Some common keywords are False, and, or, None, finally, yield, else, as, pass, or in