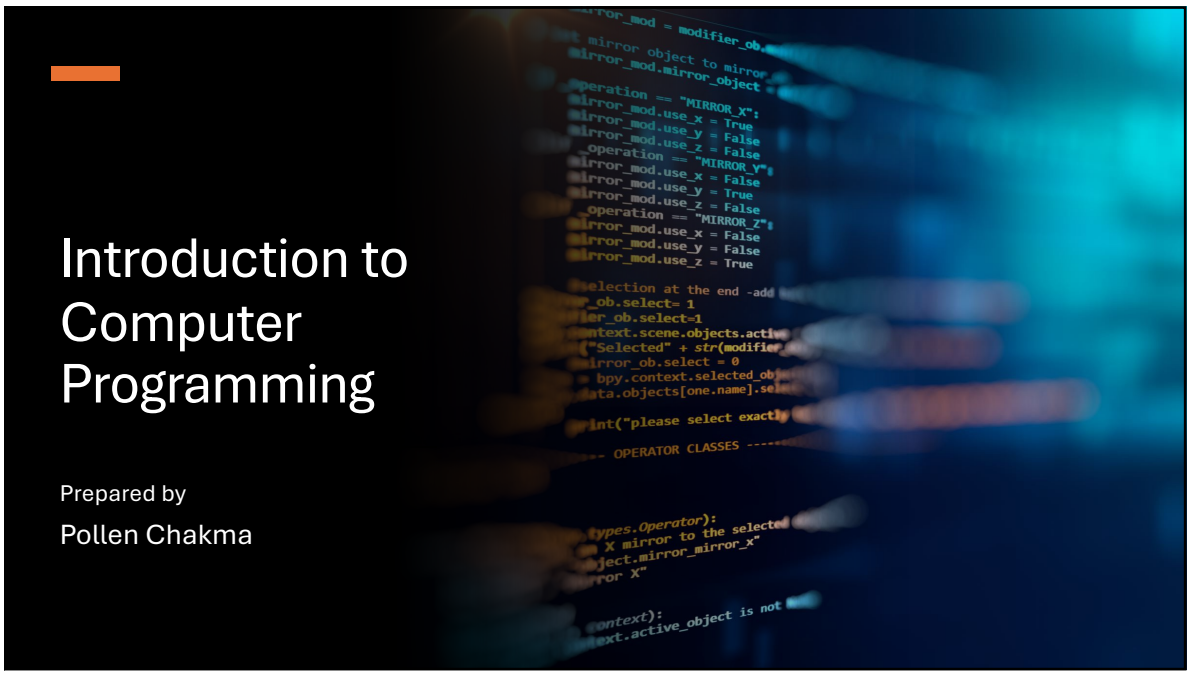
Introduction to Python

1

Introduction to Computer Programming

Prepared by
Pollen Chakma



2

Content

- Introduction to programming
- Programming and its classification
- Application of programming in Water Resources Engineering
- Why Python
- Installation
- Intro to jupyter notebook
- IDE
- Editor
- Compiler
- Debug
- Syntax(grammar)
- Variable
- Data types
- Operators

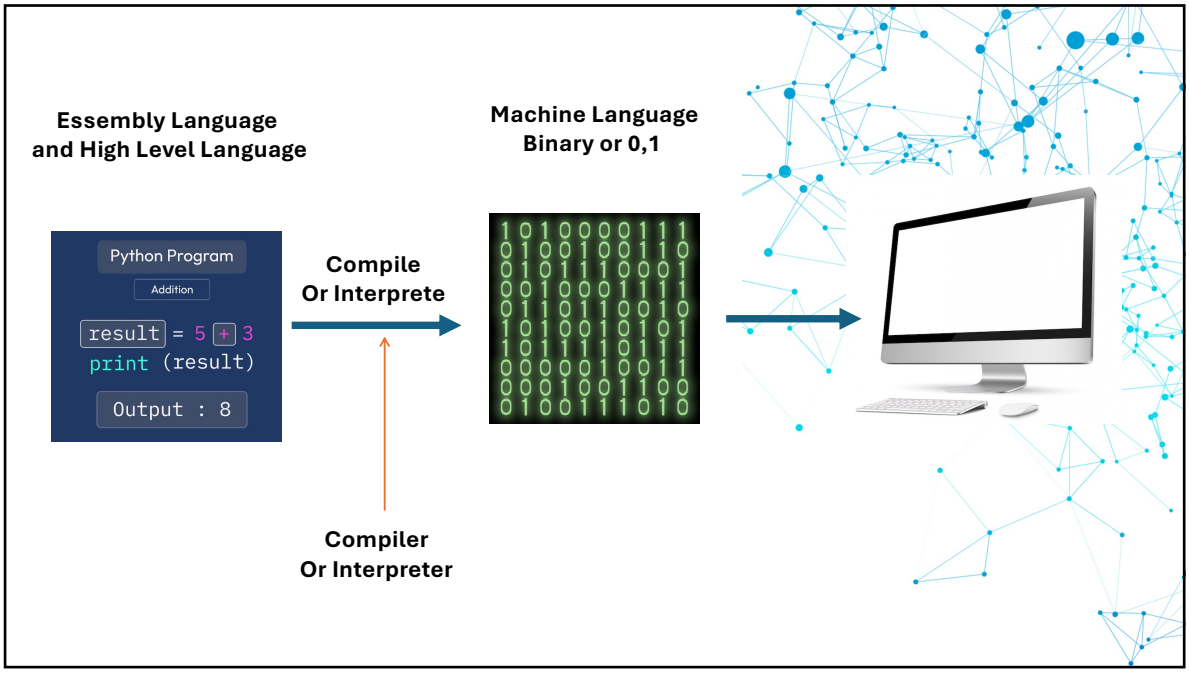
3

Programmer

Instruction Or Program

Computer can not compute or do anything by itself. Need instruction

4



5

What is a Program?

- A program is a set of instructions that tells a computer what to do.
- Examples: Open a file, Calculating numbers, sending emails, running a video game.

6

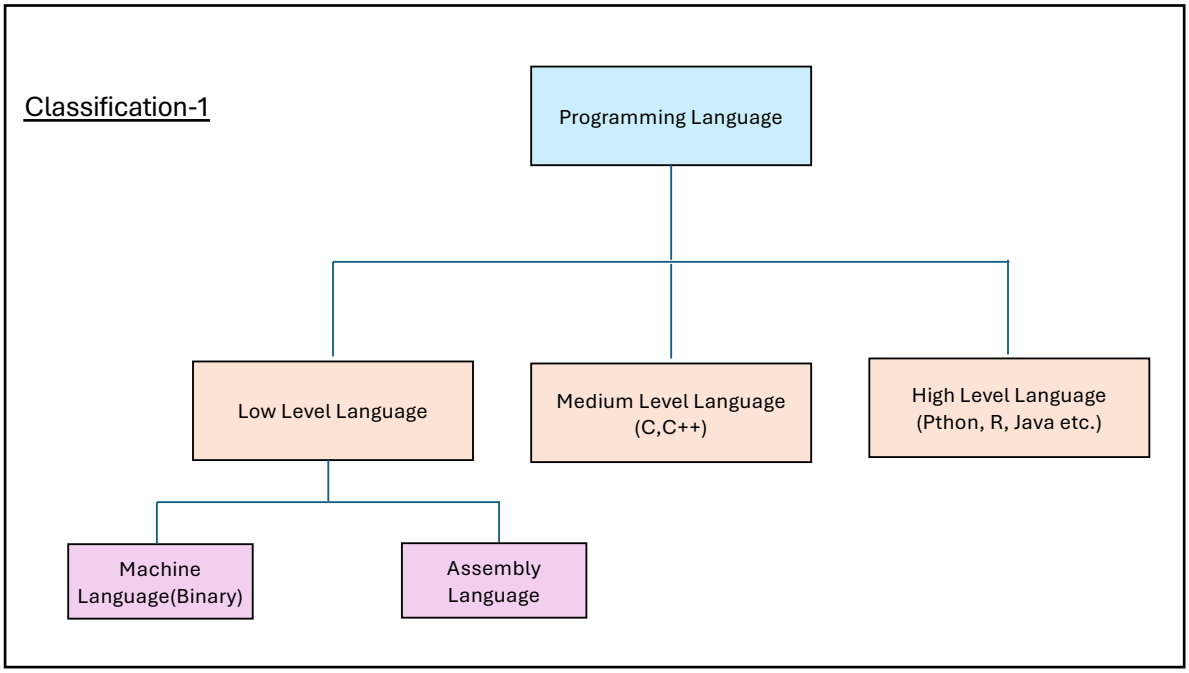
What is Programming ?

- **Programming or Computer Programming** refers to a technological process for telling a computer which tasks to perform in order to solve problems

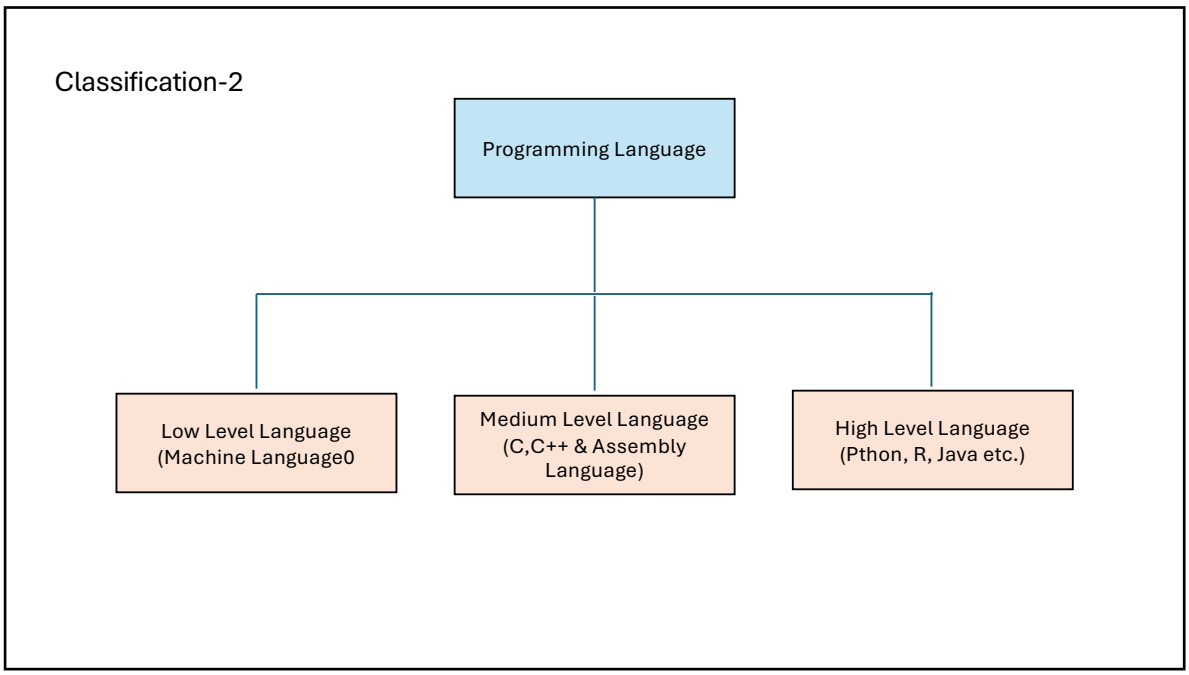
7

Classification of Programming Languages

8


Classification-1
Programming Language
Low Level Language
Medium Level Language (C,C++)
High Level Language (Pthon, R, Java etc.)
Machine Language(Binary)
Assembly Language

9


Classification-2
Programming Language
Low Level Language (Machine Language0
Medium Level Language (C,C++ & Assembly Language)
High Level Language (Pthon, R, Java etc.)

10

Compiler and Interpreter

Compiler and interpreters are two distinct methods used to translate programs written in a high level languages into machine language.

11

Compiler

A compiler processes the **entire program at once**, translating it into object code, which is usually stored in a file. This object code, also known as binary code(0,1), can be executed directly by the machine after being linked. A compiler translates the entire source code into machine code in one go. This results in **a separate executable file** that can be run independently.

Examples of compiled languages include C and C++.

12

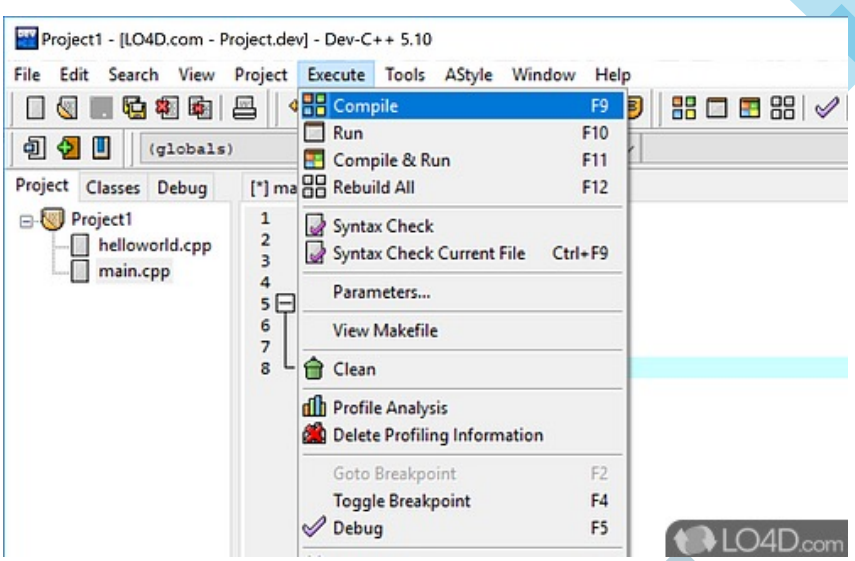


Fig: GUI of DevC++ IDE for C & C++

13

Interpreter

An interpreter **translates the source code line-by-line or statement-by-statement** into object or machine code and immediately executes it, without creating an intermediate file.

Example of interpreted languages are Python, JavaScript etc.

Thus, based on the compilation process or type of translating source code to object code programming languages can be classified as compiled and interpreted language.

14

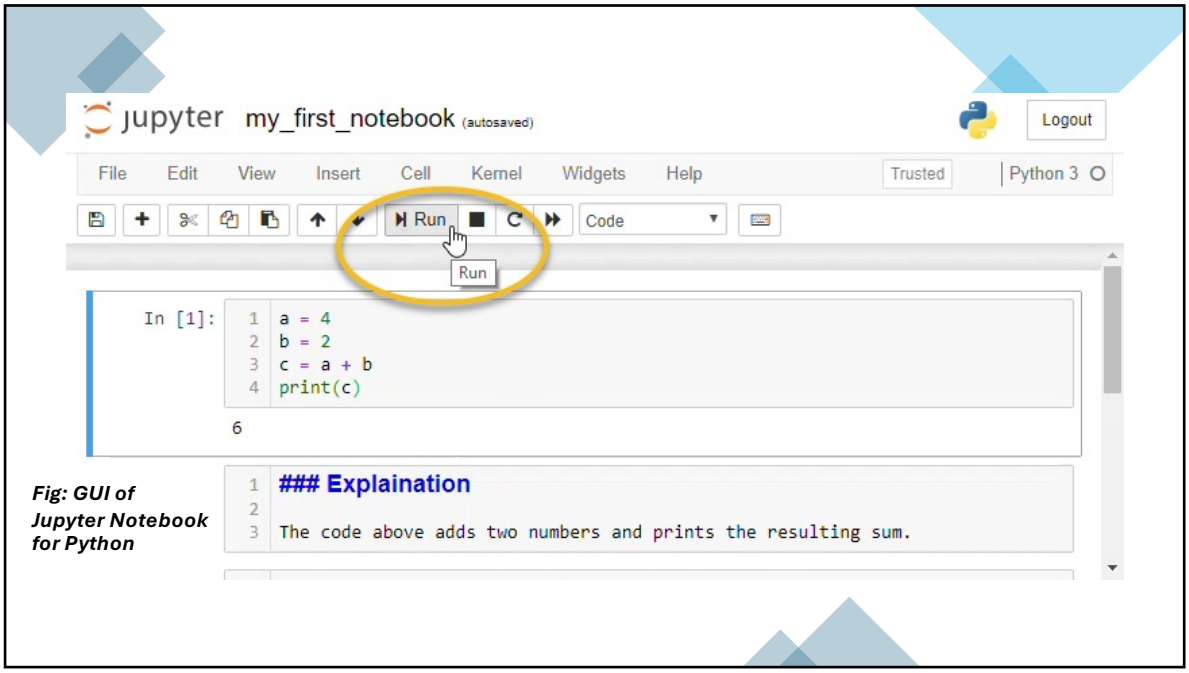


Fig: GUI of Jupyter Notebook for Python

15

Editor

- Editors or text editors are software programs that enable the user to create and edit text files. In the field of programming, the term editor usually refers to source code editors that include many special features for writing and editing code

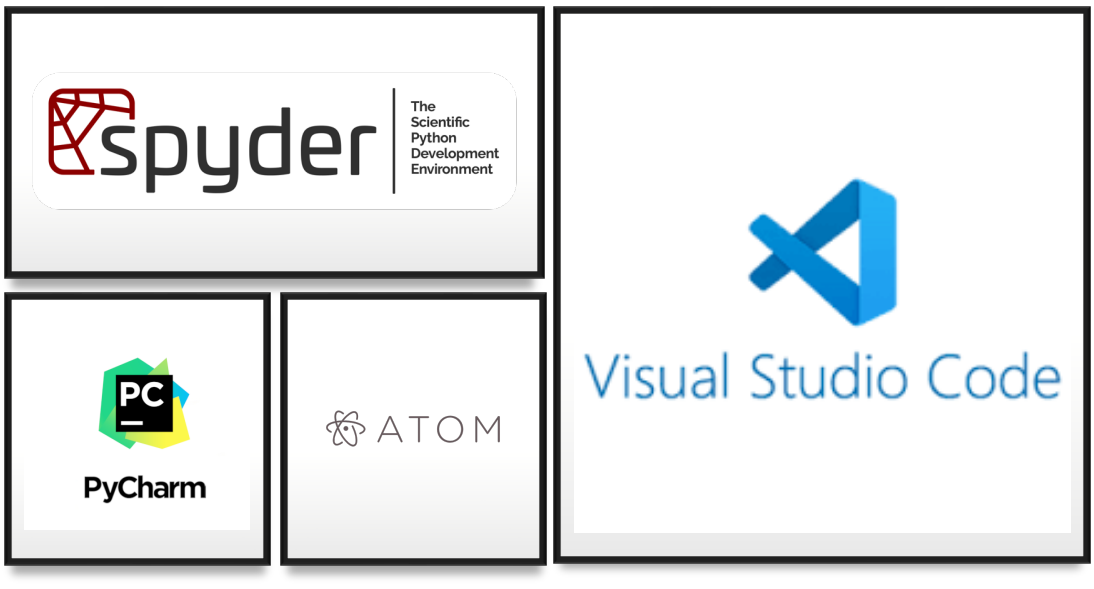
16

IDE

- An integrated development environment (IDE) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

- An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
- Compiler
- Build, execution, and debugging tools
- Some form of source control

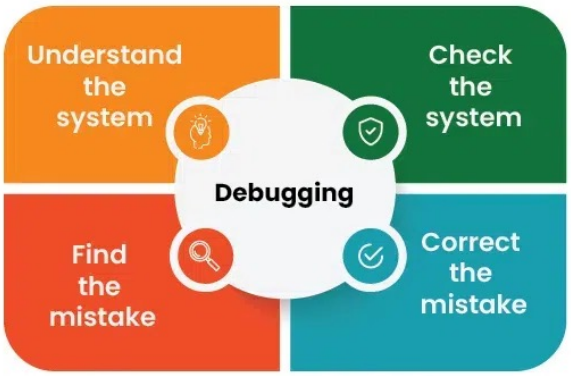
17



18

Debug

- Debugging is the process of finding and fixing errors or bugs in the source code of any software.



19

Syntax

- Programming syntax refers to the set of rules that dictate the structure and format of a programming language.

20

Variable

- A variable is a container for a value. It can be assigned a name, you can use it to refer to it later in the program.
- Based on the value assigned, the interpreter decides its data type. You can always store a different type in a variable.

```
x = 5
y = "John"
print(type(x))
print(type(y))
```

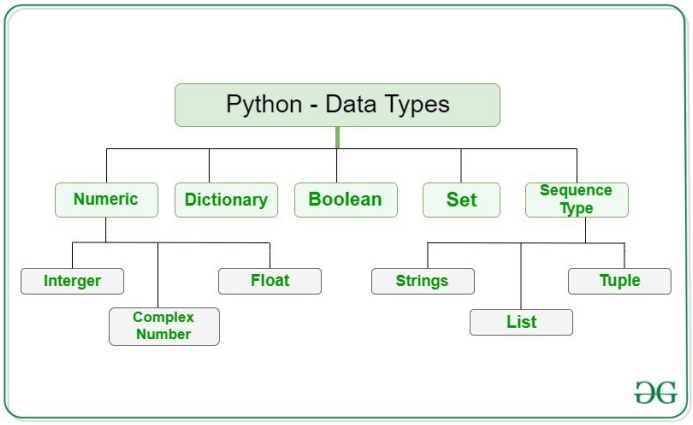
21

Python Variables

x = 45	Type = Integer 45 x
name = "DataFlair"	Type = String "DataFlair" name
nums = [1, 8.5, 9]	Type = Lists [1, 8.5, 9] nums

22

Data Type



23

Why Python

- High level language
- Simple and easy to learn
- Free and open source
- Dynamically Typed
- Extensive Library for different use case
- Usage (***Data analysis, ML,*** IoT & AI application, Desktop & web applications, games dev etc.)
- Higher Studies(MS, PHD)

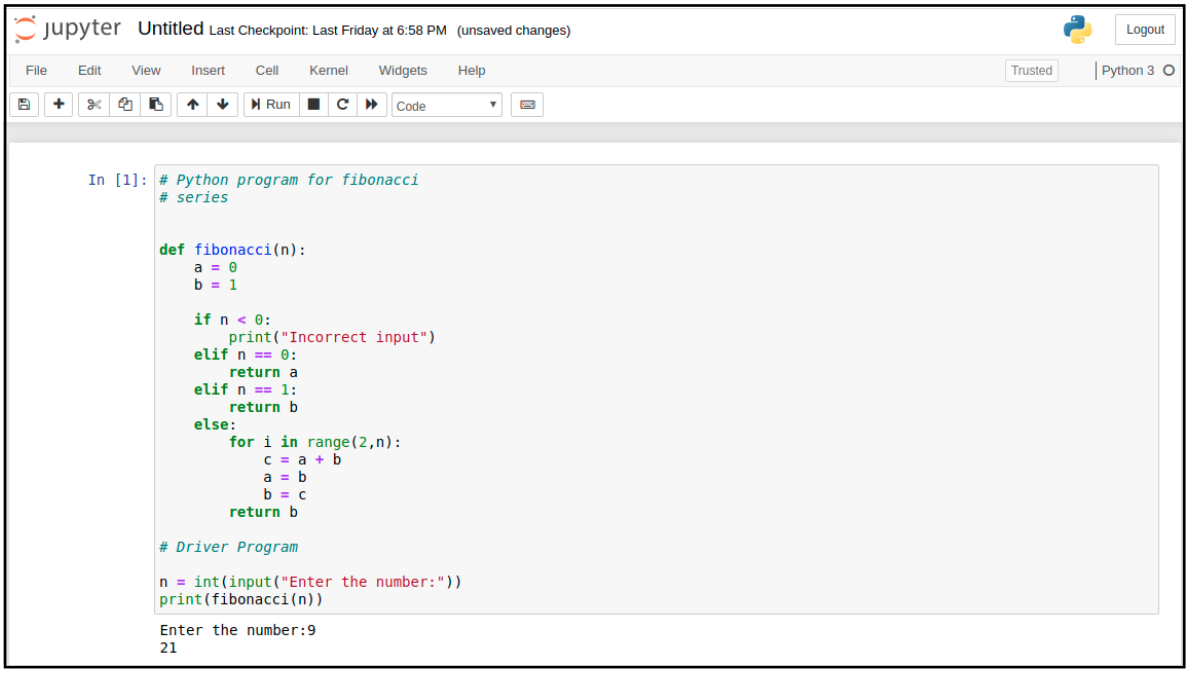
24

25

Jupyter Notebook

- Jupyter Notebook(formerly known as IPython Notebook) is a **web-based application** used to create and share interactive notebook documents, which can contain **live code, text, data visualizations, videos and other computational outputs.** Theapplication is open-source and supports the use of over 40 programming languages, including Python, R and Scala.
- Jupyter Notebook showcases real-time code results and imagery, and can execute cells in any order. This makes it a useful tool for quick code experimentation, designing code presentations or facilitating data science workflows.

26

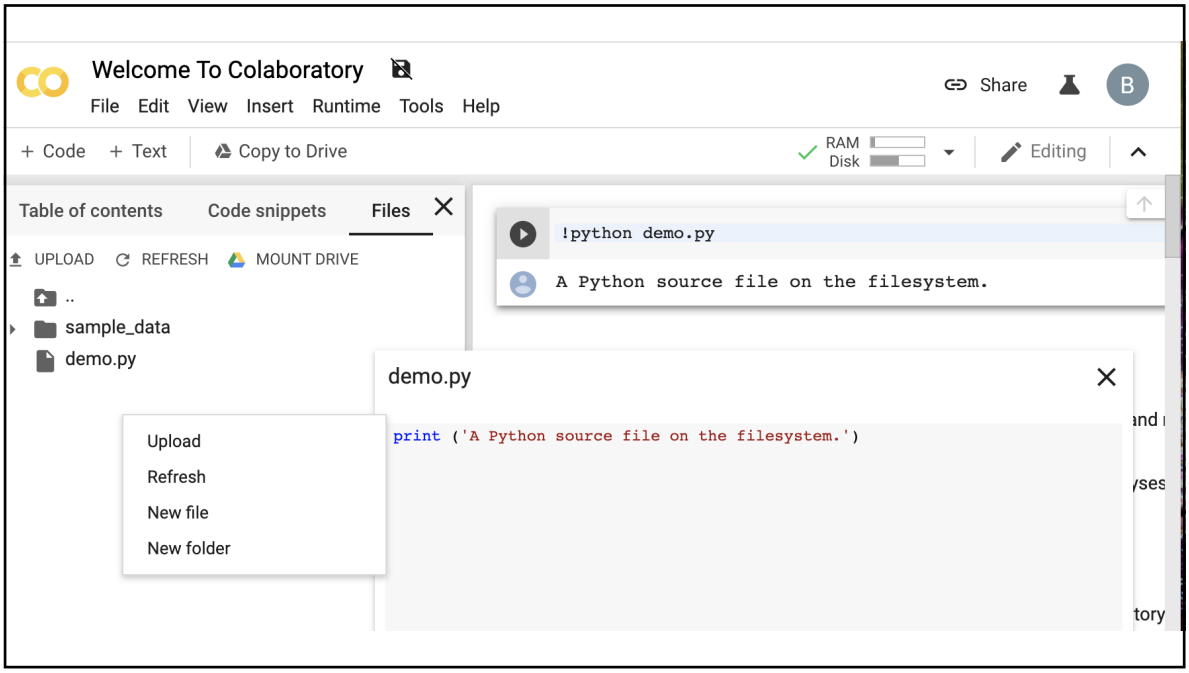


27

Google Colab

- Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup.

28



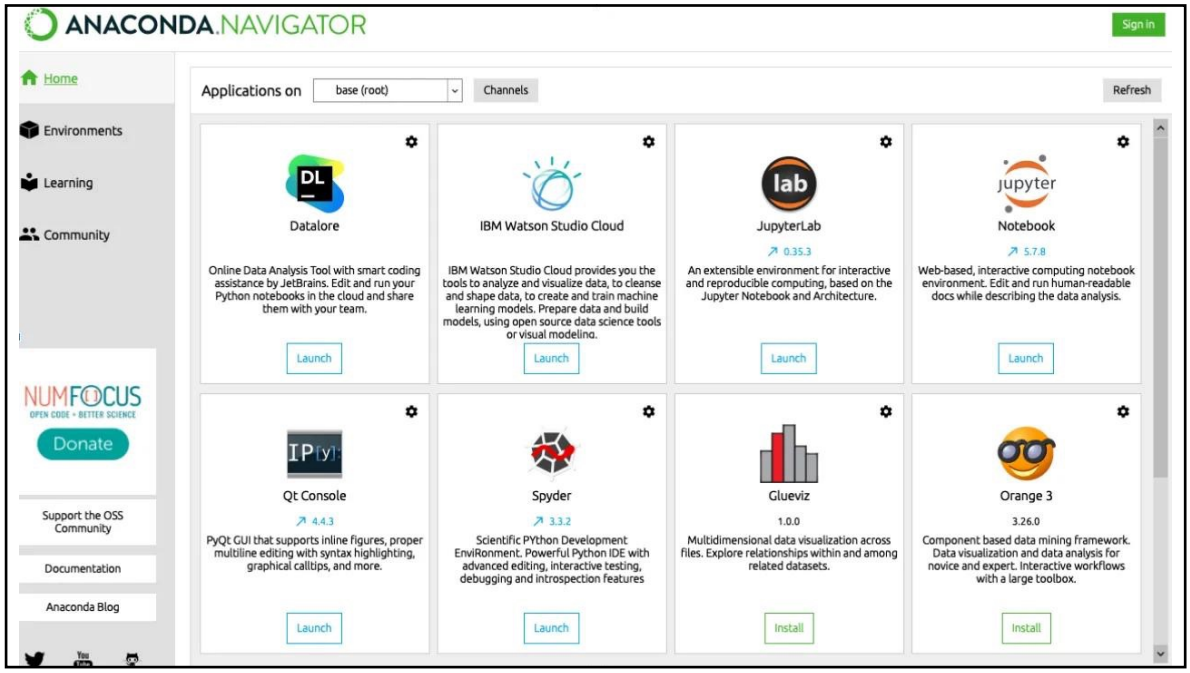
29

Anaconda Distribution

- Anaconda Distribution is a Python/R data science distribution that contains: conda - a package and environment manager

ANACONDA®

30



31

Installation Tutorial

- Anaconda installation tutotrial: https://youtu.be/WOK9HeB-OmY?si=1ZrOaLN_EmjdfdXv

32

References

- https://www.youtube.com/watch?v=ifo76VyrBYo

33

Assignment-01

Date of submission: 8/1/2025, 11:59PM

Question: *Differentiate between*

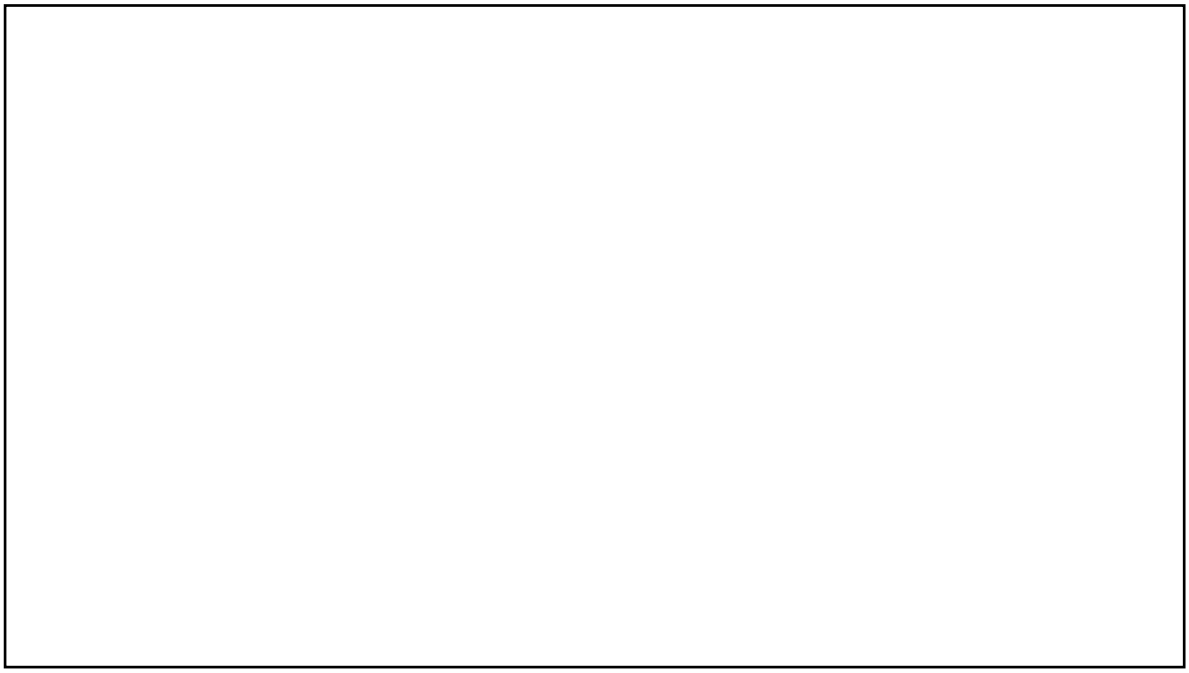
i. Low level language and high level language

ii. Machine language and assembly language

iii. Compiled language and interpreted language

Question: *Write down the advantages and disadvantages of Machine language, assembly language and high level language*

34



35