Introduction to Python

1) History of Python

Python was created by **Guido van Rossum** in the late 1980s and released in **1991**. It was designed as a successor to the ABC programming language, emphasizing readability and simplicity. Over time, Python has evolved into a powerful, multi-purpose language used in various domains, from web development to artificial intelligence.

2) History of Programming Languages

- **Binary (Machine Code):** The earliest form of programming, written in 0s and 1s, directly understood by computers.
- **Assembly Language:** A low-level language that uses mnemonic codes to represent machine-level instructions, making programming more human-readable.
- **C Language:** Developed by Dennis Ritchie in 1972 at Bell Labs, C introduced structured programming and became the foundation for many modern languages.
- **Python:** Introduced in 1991, Python brought high-level, interpreted scripting with easy-to-read syntax, making it a widely adopted language today.

3) Execution of C and Python Architecture

C Language Execution

1. Code is written in C.
2. The compiler converts it into machine code.
3. The machine executes the binary instructions.

Python Execution

1. Code is written in Python.
2. The interpreter translates it into bytecode.
3. The Python Virtual Machine (PVM) executes the bytecode.
4. If necessary, Just-In-Time (JIT) compilation optimizes performance.

4) Platform Dependent and Independent Programming Languages

- **Platform Dependent:** A language that compiles code specific to an operating system (e.g., **C** compiles differently for Windows and Linux).
- **Platform Independent:** A language that runs on multiple operating systems without modification (e.g., **Python**, as it runs via an interpreter).

5) Uses of Python

Python is widely used in various fields, including:

- **Web Development** (Django, Flask)
- **Data Science & Machine Learning** (Pandas, TensorFlow, Scikit-learn)
- **Automation & Scripting**
- **Game Development** (Pygame)
- **Cybersecurity & Ethical Hacking**
- **Embedded Systems & IoT**

Python's versatility, simplicity, and extensive libraries make it one of the most popular programming languages today.

6) Installation of Python

1. Download Python from the official website: python.org.
2. Run the installer and select **Add Python to PATH**.
3. Follow the installation prompts.
4. Verify installation by running

   ```
   python --version
   ```

   in the command prompt.

7) IDE for Python

Popular Python IDEs

- **PyCharm** - Feature-rich, best for professional developers.
- **VS Code** - Lightweight and extensible with plugins.
- **Jupyter Notebook** - Ideal for data science and ML projects.
- **IDLE** - Comes pre-installed with Python, good for beginners.
- **Spyder** - Best suited for scientific computing.

Choose an IDE based on your needs and workflow.

8) Installation of VS Code

1. Download VS Code from the official website: code.visualstudio.com.
2. Run the installer and follow the installation steps.
3. Open VS Code and install the **Python extension** from the Extensions Marketplace.
4. Configure VS Code by setting up a virtual environment or selecting the Python interpreter.
5. Verify the installation by opening a Python file and running it inside VS Code.

9) Executing First Python Query

1. Open a terminal or command prompt.
2. Type

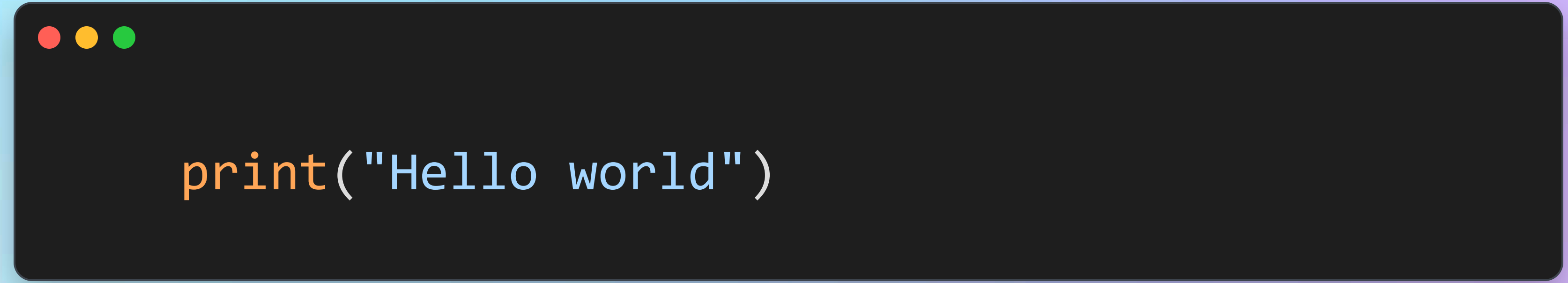
   ```
   python
   ```

   or

   ```
   python3
   ```

   to enter the Python interactive shell.

3. Run a simple command like:



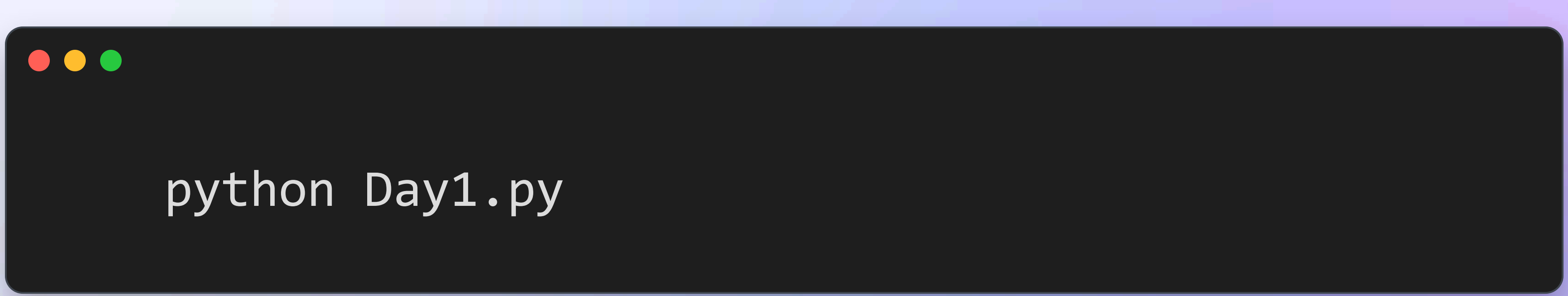
```
print("Hello world")
```

4. Press Enter to execute and see the output.

Alternatively, you can save a Python script

```
script.py
```

and run it using:



```
python Day1.py
```