Introduction to Python

BUILDING A FOUNDATION IN PYTHON FOR DATA-DRIVEN CAREERS

Python Installation

- Visit the official python https://www.python.org/downloads/
- Downloading the Python Installer

Why Python?

- **Versatility:** Used across industries for data analysis, engineering, AI, automation, and more.
- **Ease of Learning:** Simple syntax makes it beginner-friendly.
- **Community and Support:** Vast libraries and strong community support ensure help is always available.

Python in the Real World

- Data Analysis: Used by companies like Google, Netflix, and Spotify for data insights.
- Data Engineering: ETL (Extract, Transform, Load) processes in companies like Airbnb.
- Automation: Automating repetitive tasks in businesses.

Python vs. Other Data Tools

Python vs. Excel

- Handles larger datasets.
- Automates repetitive tasks with scripts.

Python vs. SQL

•SQL for database querying,

•Python for data manipulation and analysis.

Python's Key Features for Data

- **Data Structures:** Lists, dictionaries, sets, and tuples.
- **Libraries for Data:** Pandas, NumPy, Matplotlib, etc.
- **Interoperability:** Works well with SQL databases, big data tools like Hadoop, and cloud services like GCP or AWS

Introduction to Python Syntax

- Variables and Data Types.
- Basic Operations (Arithmetic, String Manipulation).

Working with Data in Python

- **Pandas for DataFrames:** Data storage, manipulation, and analysis.
- **NumPy for Numerical Data:** Efficient storage and operations on large arrays.
- **Example:** Simple data manipulation using Pandas (loading a CSV, filtering data, basic analysis).
- **Matplotlib for data Visualization:** Graphs, Plots, charts.

Python’s Role in Data Engineering

- **Data Collection:** APIs, Web Scraping.
- **Data Processing:** ETL pipelines, data cleaning, and transformation.
- **Data Storage:** Interaction with databases (SQL, NoSQL).
- **Automation:** Automating workflows and tasks with Python scripts.

Python in Data Workflow

- **Step 1:** Data Collection (APIs, Databases, Files).
- **Step 2:** Data Cleaning (handling missing values, data types).
- **Step 3:** Data Transformation (aggregations, joins).
- **Step 4:** Data Storage (save processed data to databases, files).
- **Step 5:** Data Analysis (insights, visualizations).

Python Best Practices

- Writing clean, readable code.
- Commenting and documenting your code.
- Using version control (e.g., Git).
- Testing your code.

THANK YOU