### 📘 Phase Overview

This phase lays the groundwork for everything you'll build in the field of Generative AI. Python is the de facto language for AI/ML, and mastering it — alongside tools like Git, Jupyter, and Conda — is essential.

You’ll start by learning to write clean, efficient Python code, manipulate data using essential libraries, and set up a modern development environment. You'll also gain foundational knowledge in version control and working with cloud-based notebooks for experimentation.

### 🎯 Learning Objectives

By the end of this phase, you will be able to:

* Understand and write Python programs using functions, loops, and conditionals
* Use object-oriented programming to structure code
* Work with data using NumPy and Pandas
* Visualize datasets using Matplotlib and Seaborn
* Set up and use Jupyter Notebooks, Google Colab, and VSCode
* Track and manage code with Git and GitHub
* Use virtual environments to manage dependencies

### 🧩 Module Breakdown

#### 1. Python Fundamentals (Core Language)

* Basic syntax, indentation, variables, and data types
  + int, float, str, bool, list, tuple, dict, set
* Conditional statements (if, elif, else)
* Loops (for, while), range, enumerate
* Functions and parameters, return statements
* Lambda expressions and list comprehensions
* File I/O and working with .txt, .csv
* Exception handling: try, except, finally
* Coding style: PEP8, commenting, naming conventions

📌 Hands-On Labs:

* Number guessing game
* Build a simple file parser
* Create a modular utility script

#### 2. Object-Oriented Programming (OOP)

* What is OOP and why it matters in scalable codebases
* Creating classes, methods, and objects
* \_\_init\_\_, self, class vs instance variables
* Inheritance, method overriding
* Composition and encapsulation

📌 Hands-On Labs:

* Create a class-based Bank Account system
* Implement an Inventory Manager (OOP style)

#### 3. Python for Data Manipulation

* NumPy:
  + Creating arrays, broadcasting
  + Slicing, indexing, reshaping arrays
  + Mathematical and statistical functions
* Pandas:
  + Series vs DataFrames
  + Reading/writing CSV, Excel, JSON file
  + Handling missing values, filtering, grouping, merging datasets

📌 Mini Project:

* Analyze and clean a real-world dataset (e.g., Netflix or COVID-19)

#### 4. Data Visualization

* Matplotlib:
  + Line plots, bar plots, scatter plots
  + Plot customization: labels, legends, color maps
* Seaborn:
  + Distribution plots, pair plots, heatmaps
  + Working with categorical and continuous data

📌 Mini Project:

* Visualize trends in historical stock prices
* Create a dashboard of customer behavior

#### 5. Working with Notebooks and IDEs

* Jupyter Notebooks:
  + Markdown cells, magic commands, exporting notebooks
* Google Colab:
  + GPU/TPU support, sharing, connecting to GitHub
* VSCode / PyCharm:
  + Running .py files, integrated terminal, extensions for Python

#### 6. Version Control with Git & GitHub

* Why version control matters in AI workflows
* Git basics:
  + init, clone, add, commit, push, pull, status, log
* Working with branches
* Resolving merge conflicts
* Creating pull requests
* Best practices for commits and collaboration

📌 Mini Project:

* Create a GitHub repo, push your first project
* Collaborate on a shared notebook

#### 7. Virtual Environments & Package Management

* Virtual environments: venv, conda
* Activating/deactivating environments
* Using pip to install packages
* Managing dependencies with requirements.txt or environment.yml

📌 Exercise:

* Create an isolated environment for your Python project
* Share your setup via GitHub

### 🧪 Capstone Mini-Project (End of Phase 1)

Project: *Build a Python-based EDA Toolkit*

* Read any dataset (CSV)
* Clean missing values
* Show basic stats, create visualizations
* Export report (PDF or Markdown)
* Push to GitHub as a public portfolio project