

|  |  |
| --- | --- |
| **Month 1 : Basic Python Concepts** | |
|  | |
| **Week 1 : Introduction to Python** | |
| Day 1 | Python installation and setup, basic syntax, and variables |
| Day 2 | Data types (integers, floats, strings, booleans), basic operations, and expressions |
| Day 3 | Conditional statements (if, elif, else) |
| Project | Create a command-line calculator that can perform basic arithmetic operations (addition, subtraction, multiplication, division) based on user input |
|  | |
| **Week 2 : Functions and Data Structures** | |
| Day 4 | Loops (for, while) |
| Day 5 | Introduction to functions |
| Day 6 | Advanced function concepts (map, filter) |
| Project | Implement a to-do list manager using functions to add, remove, and display tasks. Use lists to store tasks and their statuses (e.g., pending, completed) |
|  | |
| **Week 3 : Advanced Functions and File Handling** | |
| Day 7 | Lists and list operations |
| Day 8 | Tuples, sets, and dictionaries |
| Day 9 | File handling (reading and writing files) |
| Project | Develop a script that organizes files in a specified directory based on their file types (e.g., images, documents, videos). Use functions for file handling and categorization. |
|  | |
| **Week 4 : Object Oriented Programming** | |
| Day 10 | Classes and objects |
| Day 11 | Inheritance |
| Day 12 | Polymorphism |
| Project | Create classes for bank accounts with methods to deposit, withdraw, check balance, and handle exceptions (e.g., insufficient funds). Demonstrate inheritance for different account types |
|  | |

|  |  |
| --- | --- |
| **Month 2: Advanced Python Concepts** | |
|  | |
| **Week 5 : Intermediate Data Handling** | |
| Day 13 | Advanced list and dictionary comprehensions |
| Day 14 | Working with datetime and time modules |
| Day 15 | Basic debugging and testing in Python (using unittest and basic debugging techniques) |
| Project | Create a task scheduler that allows users to add tasks with due dates. The application should remind users of upcoming tasks based on the current date and time |
|  | |
| **Week 6 : Working with External Libraries** | |
| Day 16 | Introduction to NumPy |
| Day 17 | Introduction to Pandas |
| Day 18 | Data analysis and visualization with Pandas and Matplotlib |
| Project | Build a tool using NumPy and Pandas for analyzing and visualizing sample data. Perform basic statistical analysis and generate plots (e.g., histograms, scatter plots) using Matplotlib |
|  | |
| **Week 7 : Web Scraping and APIs** | |
| Day 19 | Intro to Web scraping, uses, techniques |
| Day 20 | Web scrapping with BeautifulSoup |
| Day 21 | Introduction to APIs and consuming APIs with requests |
| Project | Develop an application that retrieves weather information from a weather API using requests. Display current weather conditions and forecasts for a specified location |
|  | |
| **Week 8 : Web Development Frameworks** | |
| Day 22 | Introduction to Django |
| Day 23 | Building a simple web application |
| Day 24 | Introduction to RESTful APIs with Django |
| Project | Create a basic web blog using Django. Include features for creating, editing, and deleting blog posts. Integrate a RESTful API for accessing blog content programmatically |