**Days 1-10: Python Fundamentals & Basic Projects**

**Day 1-2: Python Setup and Basics**

* Install Python and set up your IDE (VSCode, Jupyter Notebook)
* Learn variables, data types (int, float, string, bool)
* Input/output operations (input(), print())
* [Python for Beginners (2024) - 3 Hours](https://www.youtube.com/watch?v=rfscVS0vtbw)

**Day 3-4: Control Flow**

* If-else statements, logical operators
* Loops (for, while), loop control (break, continue)
* [Control Structures in Python - 15 Minutes](https://www.youtube.com/watch?v=6iF8Xb7Z3wQ)

**Day 5-6: Functions and Modular Programming**

* Define and call functions (def keyword)
* Parameters, return values, docstrings
* [Functions in Python - 11 Minutes](https://www.youtube.com/watch?v=9Os0o3wzS_I)

**Project:** Build a **Calculator** (add, subtract, multiply, divide).

**Day 7-8: Data Structures (Lists, Tuples, Sets, Dicts)**

* Create and manipulate lists, tuples, sets, dictionaries
* List comprehensions and dictionary methods
* [Data Structures in Python - 30 Minutes](https://www.youtube.com/watch?v=R-HLU9Fl5ug)

**Project:** Develop a **To-Do List App** (add, remove, view tasks).

**Day 9-10: File Handling and Exception Management**

* Read and write files (open(), with statement)
* Error handling with try, except, finally
* [File Handling in Python - 24 Minutes](https://www.youtube.com/watch?v=Uh2ebFW8OYM)

**Project:** Create a **Contact Book** that stores and retrieves data from a file.

**Days 11-20: Object-Oriented Programming (OOP) & Intermediate Projects**

**Day 11-13: Introduction to OOP**

* Classes and objects
* Methods and attributes
* [Object-Oriented Python - 12 Minutes](https://www.youtube.com/watch?v=Ej_02ICOIgs)

**Day 14-15: Advanced OOP Concepts**

* Inheritance and polymorphism
* Encapsulation and abstraction
* [Advanced OOP in Python - 8 Minutes](https://www.youtube.com/watch?v=BJ-VvGyQxho)

**Project:** Design a **Banking System** with account classes.

**Day 16-17: Working with Modules and Packages**

* Importing modules and creating packages
* Python standard libraries (math, datetime)
* [Python Modules and Packages - 8 Minutes](https://www.youtube.com/watch?v=0oTh1CXRaQ0)

**Day 18-20: APIs and Web Scraping**

* Make HTTP requests with the requests library
* Parse HTML with BeautifulSoup
* [Web Scraping in Python - 12 Minutes](https://www.youtube.com/watch?v=ng2o98k983k)

**Project:** Build a **Weather App** that fetches data from a public API.

**Days 21-30: Data Handling & Intermediate Applications**

**Day 21-23: Data Analysis with Pandas**

* DataFrames: loading, manipulating data
* [Pandas Tutorial - 1 Hour 14 Minutes](https://www.youtube.com/watch?v=vmEHCJofslg)

**Day 24-25: Database Interaction**

* CRUD operations with SQLite
* [SQLite with Python - 27 Minutes](https://www.youtube.com/watch?v=pd-0G0MigUA)

**Project:** Create a **Student Management System** using SQLite.

**Day 26-27: Testing and Debugging**

* Unit testing with unittest
* [Python Unit Testing - 26 Minutes](https://www.youtube.com/watch?v=6tNS--WetLI)

**Day 28-30: Advanced File Operations**

* CSV, JSON handling
* [CSV and JSON in Python - 16 Minutes](https://www.youtube.com/watch?v=q5uM4VKywbA)

**Project:** Build a **Personal Diary** with timestamped entries.

**Days 31-40: Web Development with Flask**

**Day 31-33: Flask Basics**

* Set up Flask, routes, templates
* [Flask Tutorial for Beginners - 1 Hour 30 Minutes](https://www.youtube.com/watch?v=Z1RJmh_OqeA)

**Day 34-35: Forms and User Input**

* Handle forms, validate input
* [Flask Forms Tutorial - 20 Minutes](https://www.youtube.com/watch?v=2dEM-s3mRLE)

**Project:** Create a **Simple Blog** where users can add/edit posts.

**Day 36-37: Authentication**

* User registration and login
* [Flask Authentication - 30 Minutes](https://www.youtube.com/watch?v=71EU8gnZqZQ)

**Day 38-40: REST APIs with Flask**

* Build RESTful routes
* [REST APIs with Flask - 1 Hour](https://www.youtube.com/watch?v=GMppyAPbLYk)

**Project:** Develop an **API-driven Task Manager**.

**Days 41-50: Advanced Python & Automation**

**Day 41-43: Concurrency and Parallelism**

* Multithreading and multiprocessing
* [Python Concurrency & Parallelism - 30 Minutes](https://www.youtube.com/watch?v=IEEhzQoKtQU)

**Day 44-46: Data Visualization**

* Plot with Matplotlib, Seaborn
* [Data Visualization with Matplotlib - 45 Minutes](https://www.youtube.com/watch?v=G05l2c2z2KY)

**Project:** Visualize **API Request Logs**.

**Day 47-50: Automation with Python**

* Automate email sending, file handling
* [Python Automation - 1 Hour](https://www.youtube.com/watch?v=JcB6ZTAViVQ)

**Project:** Automate a **File Organizer** for directories.

**Days 51-60: Final Capstone Project & Mastery**

**Day 51-53: Planning Your Capstone**

* Design architecture and user flow

**Day 54-58: Building the Capstone**

* Implement core features

**Day 59-60: Testing and Deployment**

* Thoroughly test and document

**Capstone Project Ideas:**

* **E-commerce Platform:** Users can browse and place orders.
* **Personal Finance Tracker:** Track expenses and generate reports.
* **API Monitor:** Detect and log API anomalies in real-time.