**Week 1: Python Basics for DevOps**

**Day 1: Setting Up and Python Basics**

* **Topics**:
  + Install Python.
  + IDE setup (VS Code, PyCharm, or Jupyter Notebook).
  + Basic syntax: variables, comments, printing.
  + Data types: strings, integers, floats, lists, and dictionaries.
* **Practical**:
  + Install Python and IDE.
  + Write a script to print "Welcome to Python for DevOps."
  + Create and manipulate a list of server names.
  + Store and retrieve key-value pairs in a dictionary for server credentials.

**Day 2: Control Structures**

* **Topics**:
  + Conditional statements (if, elif, else).
  + Loops (for, while).
  + range() function.
* **Practical**:
  + Write a script to categorize server types (e.g., web, db) based on input.
  + Use a for loop to print the IP addresses from a list.
  + Implement a while loop to retry a connection until successful (simulate with a counter).

**Day 3: Functions and Modules**

* **Topics**:
  + Functions (parameters, return values).
  + Importing modules (os, sys, subprocess).
* **Practical**:
  + Create a function to check server uptime (mock output).
  + Use the os module to list all files in a directory.
  + Write a function to execute shell commands using subprocess.

**Day 4: File Handling**

* **Topics**:
  + Reading and writing files.
  + Working with configuration files.
* **Practical**:
  + Write a script to create a .txt file containing a list of server names.
  + Read and parse a log file to find errors.
  + Write a function to read and update a YAML file for configuration.

**Day 5: Error Handling**

* **Topics**:
  + try-except blocks.
  + Handling specific exceptions.
* **Practical**:
  + Write a script that connects to servers and handles errors if a server is unreachable.
  + Simulate exceptions when reading non-existent files and log errors.
  + Use a try-except block to handle invalid user inputs.

**Day 6: Working with APIs**

* **Topics**:
  + Introduction to REST APIs.
  + HTTP methods (GET, POST).
  + Using requests library.
* **Practical**:
  + Write a script to fetch data from a public API (e.g., GitHub, weather).
  + Post data (e.g., sending metrics to a monitoring tool).
  + Parse and display JSON data from the API response.

**Day 7: Automation with Python**

* **Topics**:
  + Automating tasks like backups, logs, and reports.
  + Scheduling tasks (using schedule or cron).
* **Practical**:
  + Automate a task to clean old log files from a directory.
  + Write a script to generate a daily report of server statuses.
  + Schedule the report generation using Python’s schedule library.

**Week 2: DevOps-Specific Python Skills**

**Day 8: Working with Shell Commands**

* **Topics**:
  + Execute shell commands using subprocess.
  + Parse command output.
* **Practical**:
  + Write a script to run system commands (df -h, uptime) and parse the output.
  + Automate a deployment script to restart services.

**Day 9: Networking and SSH**

* **Topics**:
  + Network programming with socket.
  + Using paramiko for SSH.
* **Practical**:
  + Write a script to check if specific ports are open on a server.
  + Use paramiko to connect to a server and execute commands.
  + Automate file transfer using SFTP (paramiko).

**Day 10: Managing Cloud Resources**

* **Topics**:
  + AWS SDK (boto3).
  + Working with instances, buckets.
* **Practical**:
  + Use boto3 to list all EC2 instances.
  + Write a script to upload files to an S3 bucket.
  + Automate stopping/starting EC2 instances.

**Day 11: Containers and Orchestration**

* **Topics**:
  + Managing Docker containers using docker-py.
  + Interacting with Kubernetes API.
* **Practical**:
  + Write a Python script to start/stop Docker containers.
  + Fetch logs of a running container.
  + Use Python to deploy a pod on a Kubernetes cluster (using kubernetes library).

**Day 12: CI/CD Pipelines**

* **Topics**:
  + Using Python in Jenkins pipelines.
  + Automating build and deploy scripts.
* **Practical**:
  + Write a Python script to trigger a Jenkins job using its API.
  + Automate a script to build a Docker image and push to Docker Hub.

**Day 13: Configuration Management**

* **Topics**:
  + Using Python with Ansible.
  + Generating dynamic inventory files.
* **Practical**:
  + Write a Python script to create dynamic inventory for Ansible.
  + Automate the generation of playbooks with variables.

**Day 14: Logging and Monitoring**

* **Topics**:
  + Integrating Python with monitoring tools.
  + Logging best practices.
* **Practical**:
  + Write a Python script to send metrics to Prometheus.
  + Automate log rotation and send alerts for critical errors.
  + Use logging module to create structured logs.