1. **Basics of Python**
2. **Control Flow**
3. **Functions and Scope**
4. **Data Structures**
5. **Classes and Objects (OOP)**
6. **Modules and Packages**
7. **Error Handling and Exceptions**
8. **File Handling**
9. **Iterators and Generators**
10. **Comprehensions**
11. **Decorators and Closures**
12. **Regular Expressions**
13. **Multithreading and Multiprocessing**
14. **Networking**
15. **Database Interaction**
16. **Data Science and Visualization (Popular Libraries)**
17. **Web Development**
18. **Testing and Debugging**
19. **Version Control with Git**
20. **Advanced Topics**
21. **Python Tools and Best Practices**

**1.Basics of Python**

* + Introduction to Python
  + Variables and Data Types
  + Input and Output (print(), input())
  + Comments and Docstrings
  + Operators (Arithmetic, Comparison, Logical, Bitwise, Assignment)

**2. Control Flow Statements**

* + Conditional Statements (if, elif, else)
  + Loops (for, while)
  + Loop Control Statements (break, continue, pass)
  + range() function

**3. Functions and Scope**

* + Defining Functions (def)
  + Function Arguments (Positional, Keyword, Default, Arbitrary \*args, \*\*kwargs)
  + Return Values (return)
  + Anonymous/Lambda Functions
  + Scope (global, nonlocal)

**4. Data Structures**

* + **Lists**: Creation, Slicing, Methods (append, pop, sort, etc.)
  + **Tuples**: Immutable sequences
  + **Dictionaries**: Key-value pairs (dict)
  + **Sets**: Unordered collections of unique items
  + **Strings**: String methods (split(), join(), replace(), etc.)

**5. Classes and Objects (OOP)**

* + Classes and Objects
  + Constructor (\_\_init\_\_)
  + Methods and Attributes
  + self keyword
  + Inheritance (Single, Multiple)
  + Method Overriding
  + Encapsulation, Polymorphism, Abstraction
  + Magic/Dunder Methods (\_\_str\_\_, \_\_repr\_\_, etc.)

**6. Modules and Packages**

* + Importing Modules (import, from ... import)
  + Built-in Modules (math, random, os, sys)
  + Creating Custom Modules
  + Python Packages (\_\_init\_\_.py file)

**7. Error Handling and Exceptions**

* + Try-Except Block
  + try-except-finally
  + Raising Exceptions (raise)
  + Custom Exceptions (class MyError(Exception))

**8. File Handling**

* + Opening and Closing Files (open(), close())
  + Reading and Writing Files (read(), write(), writelines())
  + File Modes (r, w, a, rb, etc.)
  + with statement (Context Managers)

**9. Iterators and Generators**

* + Iterators (\_\_iter\_\_, \_\_next\_\_)
  + Generators (yield keyword)
  + Generator Expressions

**10. Comprehensions**

* + List Comprehensions
  + Dictionary Comprehensions
  + Set Comprehensions

**11. Decorators and Closures**

* + Closures (Functions Inside Functions)
  + Decorators (@decorator\_name)
  + Function Wrappers (functools.wraps)

**12. Regular Expressions**

* + re Module
  + Searching (search, match, findall)
  + Replacing (sub)
  + Pattern Syntax (\d, \w, ^, $, etc.)

**13. Multithreading and Multiprocessing**

* + Threads (threading module)
  + Synchronization (Lock, Semaphore)
  + Processes (multiprocessing module)
  + Parallel Execution

**14. Networking**

* + Sockets (socket module)
  + Client-Server Communication
  + HTTP Requests (requests module)

**15. Database Interaction**

* + SQLite (sqlite3 module)
  + Connecting to Databases (CRUD operations)
  + Using SQLAlchemy or PyMySQL

**16. Data Science and Visualization (Popular Libraries)**

* + **NumPy**: Numerical operations
  + **Pandas**: Data manipulation
  + **Matplotlib**: Plotting graphs
  + **Seaborn**: Statistical data visualization
  + **Scikit-learn**: Machine Learning

**17. Web Development**

* + **Flask**: Micro web framework
  + **Django**: Full-stack web framework
  + **FastAPI**: Modern, fast web API framework

**18. Testing and Debugging**

* + unittest Module
  + pytest
  + Debugging with pdb

**19. Version Control with Git**

* + Python and Git Basics
  + Git Integration in Projects

**20. Advanced Topics**

* + Metaclasses
  + Abstract Base Classes (ABCs)
  + Context Managers (with statement)
  + Coroutines and AsyncIO (async, await)

**21. Python Tools and Best Practices**

* + Virtual Environments (venv, virtualenv)
  + Package Management (pip, pipenv, poetry)
  + Code Formatting (PEP 8, black, flake8)
  + Type Hinting (typing module)
  + Logging (logging module)



Top of Form

Bottom of Form