## ✅ **Python Topic-Wise Roadmap**

### 🟢 1. **Basics & Syntax**

* Python Installation & Setup (VS Code + Extensions)
* Running a Python program
* Variables and constants
* Comments and indentation
* input() and print() (with f-strings, .format())

### 🔁 2. **Data Types & Operators**

* Primitive Types: int, float, str, bool, NoneType
* Type Casting: str(), int(), float()
* Arithmetic Operators
* Comparison & Logical Operators: ==, !=, >, and, or
* in operator for membership testing

### 🔂 3. **Control Flow**

* Conditional Statements: if, elif, else
* Loops: for, while
* Loop Control: break, continue, pass
* Nested loops and conditions
* Iteration over strings, lists, dicts using range(), enumerate(), zip()

### 🔧 4. **Functions & Recursion**

* Defining functions using def
* Parameters & return values
* Default & keyword arguments
* Variable length arguments: \*args, \*\*kwargs
* Recursion: factorial, Fibonacci
* Lambda functions

### 🧵 5. **Strings**

* Indexing and slicing
* String methods: .lower(), .upper(), .strip(), .split(), .replace()
* String formatting: f"{}", .format()
* String comparison, counting, reversal
* Palindromes, anagrams

### 📋 6. **Lists**

* Creating & indexing lists
* List methods: append(), insert(), remove(), pop(), sort(), reverse()
* List slicing
* List comprehension: [x for x in list if x%2==0]
* Nested lists

### 🧺 7. **Tuples, Sets, Dictionaries**

* **Tuples**: immutability, indexing, unpacking
* **Sets**: uniqueness, operations – union, intersection, difference
* **Dictionaries**:
  + Key-value pairs
  + Methods: .get(), .update(), .pop(), .keys(), .values()
  + Looping through dicts

### 📁 8. **File Handling**

* open(), read(), readline(), write(), close()
* File modes: 'r', 'w', 'a', 'x'
* Using with open() (context manager)
* Writing user data to files
* Reading line count, word count

### ⚠️ 9. **Error Handling**

* try, except, finally
* Multiple exceptions
* Raising exceptions using raise
* Using assert for checks

### 📦 10. **Modules & Libraries**

* Importing Python standard modules (math, random, datetime, os)
* from module import ...
* Installing external packages with pip
* Creating your own module

### 🧱 11. **OOP (Object-Oriented Programming)**

* Creating classes and objects
* \_\_init\_\_() constructor
* Class variables vs instance variables
* Instance methods vs static/class methods
* Inheritance and method overriding
* Special methods: \_\_str\_\_(), \_\_repr\_\_()

### 🧠 12. **Advanced Topics**

* map(), filter(), reduce() with lambda
* enumerate(), zip(), list unpacking
* \*args, \*\*kwargs in-depth
* Decorators (basics only)
* Generators: yield, generator expressions

### 🧪 13. **Testing & Debugging**

* Writing simple test cases using assert
* Debugging with print() and VS Code debugger
* try/except based error logging

### 🧰 14. **APIs, JSON, and HTTP Requests**

* Parsing and creating JSON using json module
* Using requests to call APIs
* Building simple RESTful endpoints (if working with Flask)

### 🔗 15. **Bonus: Data Structures with Python**

(Helpful for DSA understanding)

* Stack using list
* Queue using collections.deque
* Heap using heapq
* HashMap using dict
* LinkedList (via class & object)