🟢 Week 1: Python Basics

Day	Topic	Tasks
1	Introduction & Setup	Install Python, IDE (VS Code/PyCharm), Write First Program
2	Variables & Data Types	Learn int, float, str, list, tuple, dict, set
3	Operators & Expressions	Arithmetic, Logical, Comparison, Membership, Identity
4	Conditional Statements	if, elif, else
5	Loops	for, while, break, continue
6	**Mini-Project**	Calculator, Number Guessing Game

🟢 Week 2: Functions & Data Structures

Day	Topic	Tasks
1	Functions	Defining, Arguments, Return Values, Lambda Functions
2	Recursion	Factorial, Fibonacci
3	Lists & Tuples	CRUD Operations, Slicing, List Comprehension
4	Dictionaries & Sets	Keys, Values, Methods, Hashing
5	String Manipulation	Slicing, Methods, Regular Expressions
6	**Mini-Project**	Contact Book, Password Generator

🟢 Week 3: File Handling & OOPs

Day	Topic	Tasks
1	File Handling	Read, Write, Append, With Statement
2	Exception Handling	try, except, finally, raise
3	Object-Oriented Programming	Class, Object, Methods
4	OOPs Concepts	Inheritance, Polymorphism, Encapsulation, Abstraction
5	Modules & Packages	Built-in & Custom Modules
6	**Mini-Project**	To-Do List, Student Management System

🟢 Week 4: Advanced Python & DSA Basics

Day	Topic	Tasks
1	Iterators & Generators	yield, next()
2	Decorators & Closures	@decorator, nested functions
3	Data Structures Intro	Lists, Stacks, Queues
4	Searching Algorithms	Linear Search, Binary Search
5	Sorting Algorithms	Bubble, Selection, Insertion
6	**Mini-Project**	Sorting Visualizer, Quiz App

🟢 Week 5: DSA (Intermediate)

Day	Topic	Tasks
1	Linked Lists	Singly, Doubly, Circular
2	Stacks	Push, Pop, Applications
3	Queues	Simple, Circular, Deque
4	Hashing	Hash Table, Collision Handling
5	Trees	Binary Tree, BST
6	**Mini-Project**	Expression Evaluator, File Organizer

🟢 Week 6: Graphs & Final Project

Day	Topic	Tasks
1	Graphs	BFS, DFS
2	Dynamic Programming	Fibonacci, Knapsack
3	Advanced Algorithms	Dijkstra, Floyd Warshall
4	Final Revision	Solve Random Problems
5	Capstone Project	Implement Final Project
6	**Final Project Submission**	Explanation & Documentation

📑 Resources to Learn From

- **Python Basics & Advanced**: Python Docs, W3Schools, Real Python
- **Data Structures & Algorithms (DSA)**: GeeksforGeeks, LeetCode, CodeChef
- **Practice Problems**: HackerRank, LeetCode, CodeWars
- **Projects**: Kaggle, PythonProjects

📌 Weekly Mini-Projects
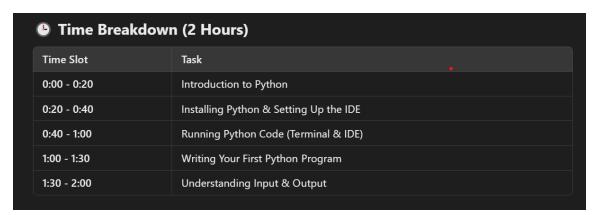
- **Week 1**: Calculator, Number Guessing Game
- **Week 2**: Contact Book, Password Generator
- **Week 3**: To-Do List, Student Management System
- **Week 4**: Sorting Visualizer, Quiz App
- **Week 5**: Expression Evaluator, File Organizer
- **Week 6**: Final Project (Choose a real-world problem)

✨ Learning Approach

1. **Concept + Code + Explain**
   - Learn the concept
   - Write and practice code
   - Try explaining to yourself or a friend
2. **Solve Problems Daily**
   - At least 5-10 questions per concept
3. **Project-Based Learning**
   - Implement mini-projects to reinforce learning

DAY of 1 learning python in 6 weeks

Where the time is splitteded as

🕐 Time Breakdown (2 Hours)

Time Slot	Task
0:00 - 0:20	Introduction to Python
0:20 - 0:40	Installing Python & Setting Up the IDE
0:40 - 1:00	Running Python Code (Terminal & IDE)
1:00 - 1:30	Writing Your First Python Program
1:30 - 2:00	Understanding Input & Output

1st introduction to python:

1 What is Python? (20 mins)

✅ **Learn:**

- Python is a high-level, interpreted programming language.
- Easy to read, write, and use for different applications (Web, AI, Data Science, Automation, etc.).
- Used by top companies like Google, Netflix, and Facebook.

✅ **Resources:**

- Python Official Docs
- W3Schools Python Intro
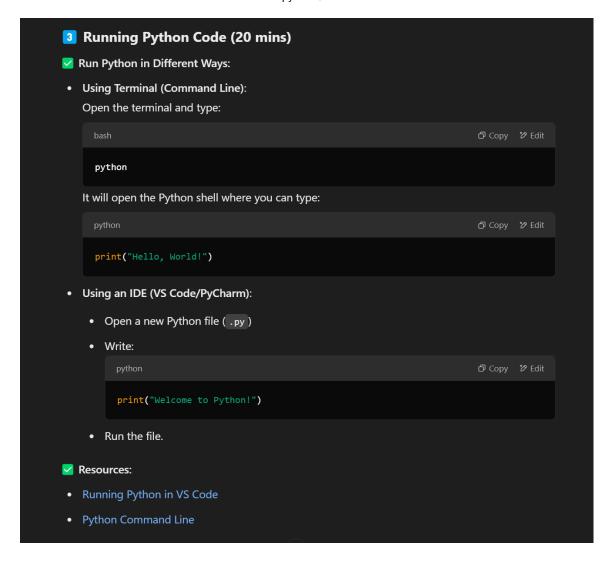
https://docs.python.org/3/tutorial/introduction.html

2nd How to install python:

2 Installing Python & IDE (20 mins)

✅ **Install Python**

- Download from Python Official Website
- Install with default settings and ensure `Add Python to PATH` is checked.

✅ **Choose an IDE** (Any one of these)

- **VS Code** (Recommended) → Download VS Code
- **PyCharm** → Download PyCharm
- **Jupyter Notebook** (For Data Science) → Install using `pip install jupyter`

✅ **Resources:**

- Python Installation Guide
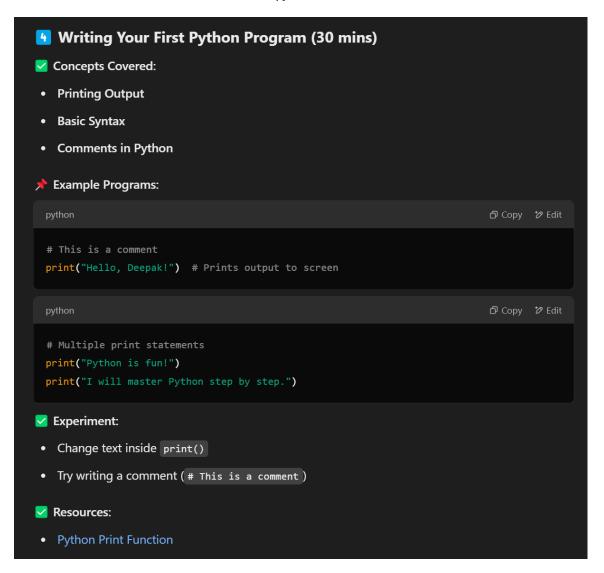
Install vs code for simple understanding:https://code.visualstudio.com/

3rd How to run code:

3️⃣ Running Python Code (20 mins)

✅ **Run Python in Different Ways:**

- **Using Terminal (Command Line):**

  Open the terminal and type:

  ```bash
  python
  ```

  It will open the Python shell where you can type:

  ```python
  print("Hello, World!")
  ```

- **Using an IDE (VS Code/PyCharm):**

  - Open a new Python file (`.py`)

  - Write:

    ```python
    print("Welcome to Python!")
    ```

  - Run the file.

✅ **Resources:**

- Running Python in VS Code

- Python Command Line
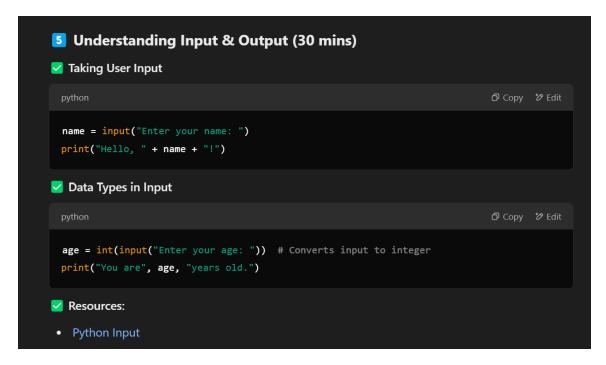
how to run code in VS code:https://code.visualstudio.com/docs/python/python-tutorial

4th how to write and execute code:

4️⃣ Writing Your First Python Program (30 mins)

✅ **Concepts Covered:**

- **Printing Output**

- **Basic Syntax**

- **Comments in Python**

📌 **Example Programs:**

```python
# This is a comment
print("Hello, Deepak!")  # Prints output to screen
```

```python
# Multiple print statements
print("Python is fun!")
print("I will master Python step by step.")
```

✅ **Experiment:**

- Change text inside `print()`

- Try writing a comment (`# This is a comment`)

✅ **Resources:**

- Python Print Function

https://www.geeksforgeeks.org/python-output-using-print-function/

5th understanding input and outputs in python:

5️⃣ Understanding Input & Output (30 mins)

✅ **Taking User Input**

```python
name = input("Enter your name: ")
print("Hello, " + name + "!")
```

✅ **Data Types in Input**

```python
age = int(input("Enter your age: "))  # Converts input to integer
print("You are", age, "years old.")
```

✅ **Resources:**

- Python Input
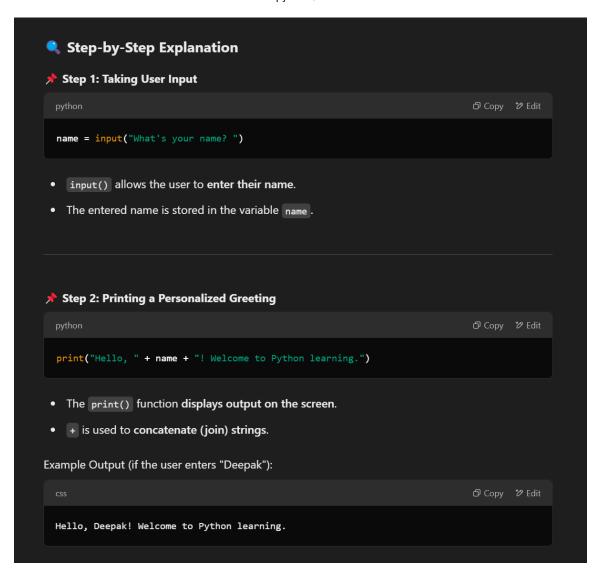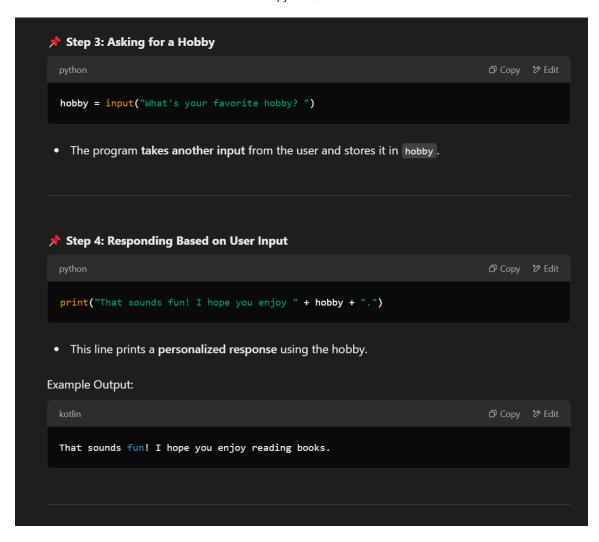
https://www.geeksforgeeks.org/input-and-output-in-python/

Task for the day:

📝 Task for Today

✅ **Hands-on Coding Tasks**

1. Print your name and age.

2. Write a program to take two numbers as input and print their sum.

3. Create a program that asks for your favorite color and prints a message using it.

✅ **Mini Project** 📌 **Simple Interactive Chatbot**

```python
name = input("What's your name? ")
print("Hello, " + name + "! Welcome to Python learning.")

hobby = input("What's your favorite hobby? ")
print("That sounds fun! I hope you enjoy " + hobby + ".")
```

Try modifying this to make it more interactive!

Ans of the project:

🖥️ Code Implementation

python   ⧉ Copy   ✎ Edit

```python
# Step 1: Ask the user for their name
name = input("What's your name? ")

# Step 2: Print a personalized greeting
print("Hello, " + name + "! Welcome to Python learning.")

# Step 3: Ask for the user's favorite hobby
hobby = input("What's your favorite hobby? ")

# Step 4: Respond with a fun message
print("That sounds fun! I hope you enjoy " + hobby + ".")

# Step 5: Ask about their learning goal
goal = input("What do you want to achieve with Python? ")

# Step 6: Provide motivation
print("That's amazing, " + name + "! Keep learning and you'll master " + goal + " in no time
```

🔍 Step-by-Step Explanation

📌 Step 1: Taking User Input

```python
name = input("What's your name? ")
```

- `input()` allows the user to **enter their name**.
- The entered name is stored in the variable `name`.

📌 Step 2: Printing a Personalized Greeting

```python
print("Hello, " + name + "! Welcome to Python learning.")
```

- The `print()` function **displays output on the screen**.
- `+` is used to **concatenate (join) strings**.

Example Output (if the user enters "Deepak"):

```css
Hello, Deepak! Welcome to Python learning.
```

📌 Step 3: Asking for a Hobby

```python
python                                                                              Copy    Edit

hobby = input("What's your favorite hobby? ")
```

- The program **takes another input** from the user and stores it in `hobby`.

📌 Step 4: Responding Based on User Input

```python
python                                                                              Copy    Edit

print("That sounds fun! I hope you enjoy " + hobby + ".")
```

- This line prints a **personalized response** using the hobby.

Example Output:

```kotlin
kotlin                                                                              Copy    Edit

That sounds fun! I hope you enjoy reading books.
```

📌 Step 5: Asking About Learning Goals

```python
goal = input("What do you want to achieve with Python? ")
```

- This asks the user what they plan to do with Python.
- The input is stored in the variable `goal`.

📌 Step 6: Motivational Message

```python
print("That's amazing, " + name + "! Keep learning and you'll master " + goal + " in no ti
```

- This provides an encouraging message based on **both name and goal.**

Example Output:

```vbnet
That's amazing, Deepak! Keep learning and you'll master Data Science in no time. 🚀
```

🎯 Example Run of the Program

```vbnet
What's your name? Deepak
Hello, Deepak! Welcome to Python learning.

What's your favorite hobby? Playing Chess
That sounds fun! I hope you enjoy Playing Chess.

What do you want to achieve with Python? Data Science
That's amazing, Deepak! Keep learning and you'll master Data Science in no time. 🚀
```

🚀 Key Python Concepts Used

- ✅ `input()` - Takes input from the user
- ✅ `print()` - Displays output
- ✅ **String concatenation** (+) - Joins strings together
- ✅ **Variables** - Stores user input

✅ Learning Outcome

By completing this mini-project, you now **understand and can explain:**

- How to take **user input**
- How to display **customized output**
- How to use **string operations**

This simple chatbot **reinforces the core Python concepts** you learned today! 💡 🚀

Summary:

📌 Summary of Day 1

- ✔ Learned what Python is
- ✔ Installed Python and IDE
- ✔ Ran Python code using different methods
- ✔ Wrote first Python programs
- ✔ Understood input and output