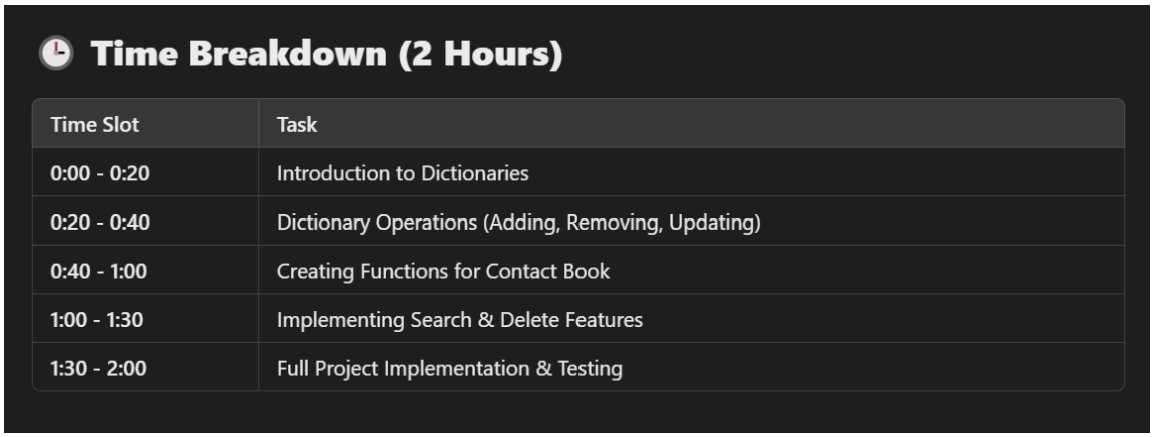
Day 6 of 6 weeks Python course:



Time Breakdown (2 Hours)

Time Slot	Task
0:00 - 0:20	Introduction to Dictionaries
0:20 - 0:40	Dictionary Operations (Adding, Removing, Updating)
0:40 - 1:00	Creating Functions for Contact Book
1:00 - 1:30	Implementing Search & Delete Features
1:30 - 2:00	Full Project Implementation & Testing

Today, you will build a Contact Book Manager using Python dictionaries. This project will reinforce your understanding of: 1.Dictionaries (key-value pairs) 2.Functions (to structure code) 3.User input handling 4.Loops & Conditional Statements By the end of today, you'll have a working contact book where users can: ✅ Add contacts ✅ Search for contacts ✅ Delete contacts ✅ View all contacts 1️⃣ What is a Dictionary? ✅ Definition: A dictionary is a collection of key-value pairs, allowing quick lookups. ✅ Syntax:

In [6]:
```
contacts = {"Alice": "9876543210", "Bob": "9123456789"}
```

✅ Accessing Values:

In [10]:
```
print(contacts["Alice"])  # Output: 9876543210
```

```
9876543210
```

✅ Dictionaries Can Store Multiple Details:

In [12]:
```
contacts = {
    "Alice": {"phone": "9876543210", "email": "alice@gmail.com"},
    "Bob": {"phone": "9123456789", "email": "bob@yahoo.com"}
}
```

✅ Adding a New Contact:

In [16]:
```
contacts["Charlie"] = "9234567890"
```

✅ Updating a Contact:

In [20]:
```
contacts["Alice"] = "9000000000"
```

✅ Removing a Contact:

In [24]:
```
del contacts["Bob"]
```

✅ Looping Through Contacts:

In [28]:
```
for name, phone in contacts.items():
    print(name, ":", phone)
```

```
Alice : 9000000000
Charlie : 9234567890
```

3️⃣ Creating Functions for Contact Book To organize code better, we'll write functions for each feature. ✅ Adding a Contact:

In [46]:
```
def add_contact(name, phone, email):
    contacts[name] = {"phone": phone, "email": email}
```

```
    print(f"{name} added successfully!")
```

✅ Viewing Contacts:

In [36]:
```
def view_contacts():
    for name, details in contacts.items():
        print(f"{name}: Phone - {details['phone']}, Email - {details['email']}")
```

4️⃣ Implementing Search & Delete Features ✅ Searching for a Contact:

In [40]:
```
def search_contact(name):
    if name in contacts:
        print(f"{name}: {contacts[name]}")
    else:
        print("Contact not found.")
```

✅ Deleting a Contact:

In [42]:
```
def delete_contact(name):
    if name in contacts:
        del contacts[name]
        print(f"{name} deleted successfully!")
    else:
        print("Contact not found.")
```

🎯 Full Mini-Project: Contact Book Manager: 💻 Complete Code Implementation:

In [1]:
```
# Contact Book Dictionary
contacts = {}

# Function to add a contact
def add_contact():
    name = input("Enter contact name: ")
    phone = input("Enter phone number: ")
    email = input("Enter email: ")
    contacts[name] = {"phone": phone, "email": email}
    print(f"{name} added successfully!\n")

# Function to view all contacts
def view_contacts():
    if contacts:
        print("\n--- Contact List ---")
        for name, details in contacts.items():
            print(f"{name}: Phone - {details['phone']}, Email - {details['email']}"
    else:
        print("No contacts available.\n")

# Function to search for a contact
def search_contact():
    name = input("Enter the name to search: ")
    if name in contacts:
        print(f"{name}: {contacts[name]}")
    else:
        print("Contact not found.\n")

# Function to delete a contact
def delete_contact():
    name = input("Enter the name to delete: ")
    if name in contacts:
```

```
        del contacts[name]
        print(f"{name} deleted successfully!\n")
    else:
        print("Contact not found.\n")

# Main Menu
while True:
    print("\n📞 Contact Book Manager 📞")
    print("1. Add Contact")
    print("2. View Contacts")
    print("3. Search Contact")
    print("4. Delete Contact")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == "1":
        add_contact()
    elif choice == "2":
        view_contacts()
    elif choice == "3":
        search_contact()
    elif choice == "4":
        delete_contact()
    elif choice == "5":
        print("Exiting Contact Book. Goodbye!")
        break
    else:
        print("Invalid choice! Please select a valid option.\n")
```

```
📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
anjali added successfully!


📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
suriya added successfully!


📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
```

```
 added successfully!


📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
Invalid choice! Please select a valid option.


📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
warner added successfully!


📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
--- Contact List ---
anjali: Phone - 9889788722, Email - anjali@gmail.com
suriya: Phone - 778899665, Email - suriya@gmail.com
: Phone - , Email -
warner: Phone - 9988556633, Email - warner@gmail.com

📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
warner: {'phone': '9988556633', 'email': 'warner@gmail.com'}

📞 Contact Book Manager 📞
1. Add Contact
2. View Contacts
3. Search Contact
4. Delete Contact
5. Exit
Exiting Contact Book. Goodbye!
```

🔍 Step-by-Step Explanation

📌 Step 1: Initializing an Empty Dictionary

python Copy Edit

```python
contacts = {}
```

- Stores all contacts.

📌 Step 2: Adding Contacts

python Copy Edit

```python
def add_contact():
    name = input("Enter contact name: ")
    phone = input("Enter phone number: ")
    email = input("Enter email: ")
    contacts[name] = {"phone": phone, "email": email}
    print(f"{name} added successfully!\n")
```

- User inputs name, phone, and email.
- Data is stored in a dictionary.

📌 Step 3: Viewing Contacts

python Copy Edit

```python
def view_contacts():
    if contacts:
        print("\n--- Contact List ---")
        for name, details in contacts.items():
            print(f"{name}: Phone - {details['phone']}, Email - {details['email']}")
    else:
        print("No contacts available.\n")
```

- Loops through the dictionary and prints all contacts.

📌 Step 4: Searching for a Contact

python Copy Edit

```python
def search_contact():
    name = input("Enter the name to search: ")
    if name in contacts:
        print(f"{name}: {contacts[name]}")
    else:
        print("Contact not found.\n")
```

- Checks if the contact exists and prints details.

📌 Step 5: Deleting a Contact

python Copy Edit

```python
def delete_contact():
    name = input("Enter the name to delete: ")
    if name in contacts:
        del contacts[name]
        print(f"{name} deleted successfully!\n")
    else:
        print("Contact not found.\n")
```

- Removes the contact from the dictionary.

📌 Step 6: Using a While Loop for Menu

python Copy Edit

```python
while True:
    print("\n📞 Contact Book Manager 📞")
    print("1. Add Contact")
    print("2. View Contacts")
    print("3. Search Contact")
    print("4. Delete Contact")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == "1":
        add_contact()
    elif choice == "2":
        view_contacts()
    elif choice == "3":
        search_contact()
    elif choice == "4":
        delete_contact()
    elif choice == "5":
        print("Exiting Contact Book. Goodbye!")
        break
    else:
        print("Invalid choice! Please select a valid option.\n")
```

- Loops until the user selects **Exit (5)**.

📌 Summary of Day 6 ✔ Learned Dictionaries & Operations ✔ Implemented Contact Book Features ✔ Completed a Mini-Project: Contact Book Manager