Day 5 of 6 weeks Python course:

Time Breakdown (2 Hours)

Time Slot	Task
0:00 - 0:20	Introduction to Lists
0:20 - 0:40	List Operations & Methods
0:40 - 1:00	Introduction to Tuples
1:00 - 1:20	Tuple Operations & Differences from Lists
1:20 - 1:30	List vs. Tuple: When to Use What?
1:30 - 2:00	Mini-Project: Student Marks Analyzer

1 What is a List? ✅ Definition: A list is a mutable (changeable) collection of items in Python. ✅ Syntax:

In [4]:
```
my_list = [1, 2, 3, 4, 5]
```

✅ Example Usage:

In [6]:
```
fruits = ["apple", "banana", "cherry"]
print(fruits)  # Output: ['apple', 'banana', 'cherry']
```

```
['apple', 'banana', 'cherry']
```

✅ Lists Can Contain Different Data Types

In [8]:
```
mixed_list = [10, "Python", 3.14, True]
print(mixed_list)
```

```
[10, 'Python', 3.14, True]
```

✅ Accessing List Elements:

In [10]:
```
print(fruits[0])  # First element: 'apple'
print(fruits[-1])  # Last element: 'cherry'
```

```
apple
cherry
```

2 List Operations & Methods ✅ Basic List Operations:

In [12]:
```
numbers = [1, 2, 3, 4, 5]

numbers.append(6)  # Adds 6 at the end
numbers.insert(2, 10)  # Inserts 10 at index 2
numbers.remove(3)  # Removes first occurrence of 3
numbers.pop()  # Removes last element
numbers.sort()  # Sorts the list
numbers.reverse()  # Reverses the list

print(numbers)
```

```
[10, 5, 4, 2, 1]
```

✅ List Slicing:

In [14]:
```
nums = [0, 1, 2, 3, 4, 5]
print(nums[1:4])  # Output: [1, 2, 3]
```

```
print(nums[:3])  # Output: [0, 1, 2]
print(nums[-3:])  # Output: [3, 4, 5]
```

```
[1, 2, 3]
[0, 1, 2]
[3, 4, 5]
```

✅ Looping Through Lists:

In [16]:
```
for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
```

3️⃣ What is a Tuple? ✅ Definition: A tuple is an immutable (unchangeable) collection of items. ✅ Syntax:

In [20]:
```
my_tuple = (1, 2, 3, 4, 5)
```

✅ Example Usage:

In [26]:
```
colors = ("red", "green", "blue")
print(colors[0])  # Output: 'red'
```

```
red
```

✅ Why Use Tuples? Faster than lists (since they are immutable) Useful when you want fixed data (e.g., coordinates, database records) 4️⃣ Tuple Operations & Differences from Lists ✅ Basic Tuple Operations:

In [32]:
```
numbers = (1, 2, 3, 4, 5)

print(numbers[1])  # Access element
print(len(numbers))  # Get tuple length

# Tuples are immutable, so these will cause an error:
# numbers.append(6) ❌ Not allowed
# numbers[0] = 10 ❌ Not allowed
```

```
2
5
```

✅ Looping Through a Tuple

In [34]:
```
for color in colors:
    print(color)
```

```
red
green
blue
```

✅ Tuple Unpacking

In [37]:
```
x, y, z = colors
print(x, y, z)  # Output: red green blue
```

```
red green blue
```

5️⃣ List vs. Tuple: When to Use What? Feature List Tuple Mutability ✅ Mutable (Can be changed) ❌ Immutable (Cannot be changed) Performance Slower (Uses more memory) Faster (Uses less memory) Use Case When data changes frequently When data is fixed (e.g., coordinates, database records) ✅ When to use a List? When you need modification (e.g., to-do lists, inventories). ✅ When to use a Tuple? When data should not change (e.g., days of the week, fixed settings).Mini-Project: Student Marks Analyzer 📌 Project Goal Ask the user for marks in multiple subjects . Store marks in a list. Calculate the average, highest, and lowest marks. Display results.

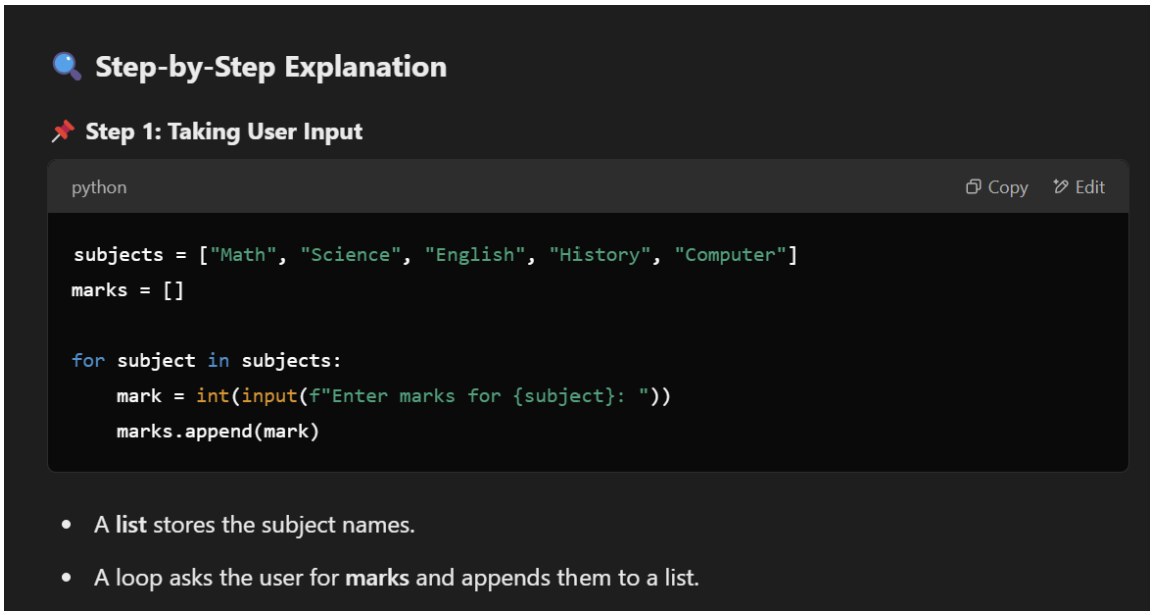
In [43]:
```python
# Step 1: Get marks from user
subjects = ["Math", "Science", "English", "History", "Computer"]
marks = []

for subject in subjects:
    mark = int(input(f"Enter marks for {subject}: "))
    marks.append(mark)

# Step 2: Calculate statistics
average = sum(marks) / len(marks)
highest = max(marks)
lowest = min(marks)

# Step 3: Display results
print("\n--- Student Marks Report ---")
print("Subjects:", subjects)
print("Marks:", marks)
print(f"Average Marks: {average:.2f}")
print(f"Highest Marks: {highest}")
print(f"Lowest Marks: {lowest}")
```

```
--- Student Marks Report ---
Subjects: ['Math', 'Science', 'English', 'History', 'Computer']
Marks: [86, 89, 94, 78, 89]
Average Marks: 87.20
Highest Marks: 94
Lowest Marks: 78
```



📌 Step 2: Calculating Statistics

python Copy Edit

```python
average = sum(marks) / len(marks)
highest = max(marks)
lowest = min(marks)
```

- `sum(marks) / len(marks)` calculates **average marks.**
- `max(marks)` finds the **highest mark.**
- `min(marks)` finds the **lowest mark.**

📌 Step 3: Displaying Results

python Copy Edit

```python
print("\n--- Student Marks Report ---")
print("Subjects:", subjects)
print("Marks:", marks)
print(f"Average Marks: {average:.2f}")
print(f"Highest Marks: {highest}")
print(f"Lowest Marks: {lowest}")
```

- The program **displays the results** in a structured format.

✅ Example Output:

rust Copy Edit

```rust
Enter marks for Math: 90
Enter marks for Science: 85
Enter marks for English: 78
Enter marks for History: 88
Enter marks for Computer: 95

--- Student Marks Report ---
Subjects: ['Math', 'Science', 'English', 'History', 'Computer']
Marks: [90, 85, 78, 88, 95]
Average Marks: 87.20
Highest Marks: 95
Lowest Marks: 78
```

📌 Summary of Day 5 ✔ Learned Lists & Tuples ✔ Practiced List & Tuple Operations ✔ Completed a Mini-Project: Student Marks Analyzer