🧮 Assignment: Assignment Operators in Python

🎯 **Objective:**
To understand and apply Python's assignment operators (`=`, `+=`, `-=`, `*=`, `/=`, `//=`, `%=`, `**=`) through practical coding tasks.

🟢 **Level 1 – Easy**

✅ **Task 1: Basic Assignment Practice**

Write a Python program that:

- Declares a variable `x = 10`
- Perform and display results of the following operations:
  - `x += 5`
  - `x -= 3`
  - `x *= 2`
  - `x /= 4`
  - `x %= 3`
  - `x **= 2`
  - `x //= 3`

✔️ *Focus: Syntax and understanding of how each operator modifies the variable.*

🟡 **Level 2 – Medium**

✅ **Task 2: Wallet Balance Simulation**

- Initialize a wallet balance as ₹1000.
- Perform the following transactions using assignment operators:
  1. Add salary of ₹5000
  2. Deduct grocery expense of ₹1200
  3. Deduct utility bill of ₹800
  4. Add cashback of ₹200
  5. Apply 10% savings using `*=` or `/=`

✔️ *Focus: Real-world logic with chained assignment updates.*

🧮 Assignment: Assignment Operators in Python

✅ Task 3: Compound Growth

- Start with `investment = 1000`
- Assume it grows 5% annually for 3 years.
- Use `*= 1.05` in a loop to update the value.

✔️ *Focus: Repeated updates with `*=` operator*

🔴 Level 3 – Hard

✅ Task 4: Dynamic Score Tracker (Game Scenario)

- Initialize a score variable to 0.
- Simulate a game where:
  - Player collects 10 points → `+=`
  - Player hits a trap and loses 5 points → `-=`
  - Player finds a multiplier bonus and score doubles → `*=`
  - Final score is floored to nearest 10 → `//=`

✔️ *Focus: Sequence of multiple assignment updates mimicking a state change.*

✅ Task 5: Mathematical Pattern Generator

- Input a number `n`.
- Use assignment operators inside a loop to:
  - Build a geometric progression.
  - For example: `result *= 2` on each iteration.
- Print the first 5 powers of 2 (2, 4, 8, 16, 32).

✔️ *Focus: Repeated application of `*=` and display patterns.*