Q1. The Variable Swap

Goal: Assign two numbers to variables and then swap their values.

- Assign `a = 10` and `b = 20`.
- Without simply typing `a = 20`, find a way to make `a` hold the value of `b` and vice versa.
- **Hint:** You might need a third variable called `temp` to hold a value for a moment!

Q2. The Equality Checker

Goal: Understand the difference between assigning a value and checking a value.

- Create a variable `secret_number` and set it to `7`.
- Write a line of code that checks if `secret_number` is equal to `10`.
- **Expected output:** The console should return `False`.

Q3. Calculating a Total

Goal: Use the equals sign to store the result of a math operation.

- Create three variables: `apple_price = 1.50`, `banana_price = 0.75`, and `quantity = 4`.
- Create a fourth variable called `total_cost`.
- Use the equals sign to assign `total_cost` the value of `(apple_price + banana_price) * quantity`.

Q4. String Comparison

Goal: See how Python handles "equal to" with text.

- Assign `password_stored = "Python123"`.
- Assign `password_entered = "python123"`.
- Write a comparison to see if `password_stored == password_entered`.
- **Think about this:** Why does Python say these are **not** equal?

Q5. The Boolean Assignment

Goal: Save the result of a logic test into a new variable.

- Set `age = 15`.
- Create a variable called `can_drive`.
- Assign `can_drive` the result of the comparison `age >= 16`.
- Print `can_drive` to see if the result is `True` or `False`

Q6. The "Update" Shortcut

Goal: Practice the `+=` operator to update a score.

- Assign `score = 50`.
- A player just earned 15 bonus points. Instead of writing `score = score + 15`, use the **addition assignment** operator (`+=`) to update the variable.
- Print the new `score`.

Q7. Comparing Expressions

Goal: See how Python evaluates math on both sides of an equality.

- Write a single line of code to check if 10^2 (written as `10 ** 2`) is equal to 50×2.
- **Challenge:** Can you guess if the result will be `True` or `False` before you run it?

Q8. The Type Trap

Goal: Understand how data types affect equality.

- Assign `x = 5` (an integer).
- Assign `y = "5"` (a string/text).
- Check if `x == y`.
- **Observation:** Even though they look like the same number, does Python consider them equal?

Q9. Multi-Variable Assignment

Goal: Learn the Python shortcut for assigning multiple variables at once.

- In a single line of code, assign the value `1` to `red`, `2` to `green`, and `3` to `blue`.
- **Hint:** It looks like this: `var1, var2, var3 = val1, val2, val3`.

Q10. The Budget Checker

Goal: Combine math and equality for a real-world scenario.

- Assign `budget = 100`.
- Assign `item_1 = 45` and `item_2 = 55`.
- Create a variable called `is_exact_change`.
- Assign `is_exact_change` the result of checking if `item_1 + item_2` is exactly equal to `budget`.
-

Q11.The Coffee Shop Point-of-Sale

The Setup: You are building a simple cash register for a local cafe. You need to store the price of a latte, the quantity ordered, and the tax rate to calculate the final total.

Your Tasks:

1. Create a variable `item_name` and set it to `"Latte"`.
2. Create `price` (float) and `quantity` (integer).
3. Create a `tax_rate` variable set to **0.08** (8%).
4. Calculate the `subtotal` (price times quantity).
5. Calculate the `total_price` including tax.
6. **Print** a receipt sentence like: *"Your 3 Lattes will cost $16.20."*

Q12: The Space Explorer's Dashboard

The Setup: You're coding the HUD (Heads-Up Display) for a spacecraft. Variables here need to be updated frequently as the "state" of the ship changes.

Your Tasks:

1. Initialize `fuel_level` at **100**.
2. Initialize `is_engine_on` as a boolean (`True` or `False`).
3. **Update the variable:** The ship just took off! Decrease the `fuel_level` by **15**.
4. **Re-assignment:** Change `is_engine_on` to the opposite of its current state.
5. Create a list variable called `destination_planets` containing three strings (e.g., "Mars", "Jupiter", "Saturn").

Q13: The Social Media Profile

The Setup: You are managing user data for a new social media app. This involves handling different data types and "f-strings" for output.

Your Tasks:

1. Store a user's `username`, `age`, and `follower_count`.
2. Create a boolean variable `is_private_account`.
3. **The "Follow" Logic:** A new person followed the user. Increment the `follower_count` by **1** using the `+=` operator.
4. **Profile Update:** The user changed their username. Re-assign the `username` variable to a new string.
5. Print a summary of the profile using an f-string