🐣 BEGINNER PYTHON PROJECTS (1–10)

1. Hello World App

🎯 **Goal:** Display “Hello, World!” and ask for the user’s name.
⚙️ **Steps:**

1. Use `print()` to display messages.
2. Use `input()` to ask the user’s name.
3. Print a greeting like: “Hello, Nervely!”
   📁 **File name:** `hello_world.py`
   🚫 **Restrictions:**

- No fancy imports.
- Only use `print()` and `input()`.
- Keep it under 10 lines.

2. Simple Calculator

🎯 **Goal:** Perform addition, subtraction, multiplication, and division.
⚙️ **Steps:**

1. Ask for two numbers.
2. Ask the user to choose an operation (+, -, *, /).
3. Show the result.
   📁 **File name:** `calculator.py`
   🚫 **Restrictions:**

- Use `if`/`elif` for operations.
- Handle division by zero (`if b == 0:`).
- No external modules.

3. Guess the Number Game

🎯 **Goal:** The computer picks a random number; you try to guess it.
⚙️ **Steps:**

1. Import `random`.
2. Generate a random number between 1 and 20.
3. Let the user guess until they find the number.

4. Tell if the guess is too high or too low.
   📁 **File name:** `guess_number.py`
   🚫 **Restrictions:**

- Don't show the number to the user 😅.
- Use `while` loop and `break`.

4. Even or Odd Checker

🎯 **Goal:** Tell if a number is even or odd.
⚙️ **Steps:**

1. Ask the user for a number.
2. Use % (modulus) to check remainder.
3. Print if it's even or odd.
   📁 **File name:** `even_odd.py`
   🚫 **Restrictions:**

- Only 1 condition (`if/else`).
- Handle negative numbers too.

5. Temperature Converter

🎯 **Goal:** Convert Celsius ↔ Fahrenheit.
⚙️ **Steps:**

1. Ask the user what they want to convert (C or F).
2. Ask for the temperature.
3. Apply the formula:
   - °F = (°C × 9/5) + 32
   - °C = (°F − 32) × 5/9
4. Show result.
   📁 **File name:** `temp_converter.py`
   🚫 **Restrictions:**

- Use float, not int.
- Round result to 2 decimals (`round(value, 2)`).

6. Simple Password Generator

🎯 **Goal:** Create random passwords.
⚙️ **Steps:**

1. Import `random` and `string`.
2. Ask how long the password should be.
3. Use `string.ascii_letters + string.digits + string.punctuation`.
4. Generate and show the password.
   📁 **File name:** `password_generator.py`
   🚫 **Restrictions:**

- Don't print the password before generating it.
- No hardcoded passwords.

7. To-Do List (Console Version)

🎯 **Goal:** Add, show, and remove tasks.
⚙️ **Steps:**

1. Create an empty list `tasks = []`.
2. Use a loop:
   - Add a task
   - View tasks
   - Remove a task
   - Exit
3. Save tasks temporarily (no file yet).
   📁 **File name:** `todo_list.py`
   🚫 **Restrictions:**

- Don't use databases or files yet.
- Use `while True` and break when user types "exit".

8. Countdown Timer

🎯 **Goal:** Countdown from a number of seconds to 0.
⚙️ **Steps:**

1. Import `time`.
2. Ask for the number of seconds.
3. Use a loop and `time.sleep(1)`.
4. Print each second.
   📁 **File name:** `countdown_timer.py`
   🚫 **Restrictions:**

- Use only `while` or `for` loop.
- Don’t skip seconds.

9. Simple Alarm Clock

🎯 **Goal:** Play a sound or message at a set time.
⚙️ **Steps:**

1. Ask for the target time (HH:MM).
2. Use `datetime` to check the current time.
3. Wait until the time matches, then print “Time’s up!”.
   📁 **File name:** `alarm_clock.py`
   🚫 **Restrictions:**

- No `playsound` or music yet (just print).
- Use `while True`.

10. Word Counter

🎯 **Goal:** Count how many words are in a sentence or text.
⚙️ **Steps:**

1. Ask the user to enter a text.
2. Split it with `.split()`.
3. Count the number of words using `len()`.
   📁 **File name:** `word_counter.py`
   🚫 **Restrictions:**

- Don’t use external libraries.
- Ignore extra spaces (use `.strip()`).