Mad labs python program explanation

- **Prompt for Input**:
- The function asks the user for input using the `input(prompt)` function.
- `.strip()` removes any leading or trailing spaces from the user's input.
- **Check for Non-Empty Input**:
- The `if user_input:` line checks if the input is **not empty**. If the user has entered something, the input will be returned.
- If the input is empty (i.e., the user didn't enter anything or just typed spaces), the program prints a message and repeats the prompt.
- **Loop Until Valid Input**:
- The `while True:` loop makes sure the program keeps asking for input until the user provides a valid, non-empty answer.

2. **Randomly Select a Story Template**

Explanation:

- The function **`random.choice(story_templates)`** randomly selects one of the predefined templates from the list `story_templates`. Each time the program runs, this will give the user a different experience.

Why it's important:

- **Randomization** ensures that each playthrough is different, and users get a fresh story every time, enhancing replayability and fun.

3. **Get User Inputs**

Explanation:

- We prompt the user to provide specific types of words, like adjectives, nouns, and verbs. Each time, the program calls the `get_input()` function, which handles asking the user for input.
- The user is asked to fill in the placeholders in the story template:
    - **Adjective**: Describes a noun (e.g., "happy," "silly").
    - **Nouns**: A person, place, or thing (e.g., "dog," "park").
    - **Verbs**: An action word, where `verb1` is past tense (e.g., "ran"), and `verb2` is present tense (e.g., "jump").
    - **Place**: A location (e.g., "school," "city").

4. **Get Input with Validation (The `get_input()` function)**

Explanation:

- **`get_input(prompt)`** is a helper function that asks for input and ensures the user doesn't submit empty or just space input.
    - The function keeps asking the user for input in a loop (`while True`) until a valid (non-empty) response is given.
    - The `.strip()` method is used to remove leading and trailing spaces. This means that if the user enters spaces (e.g., `"  "`), they are ignored.

Explanation:

- **`.format()`** is used to insert the values of the user's input into the selected template.
    - Each placeholder in the template (e.g., `{adjective}`, `{noun1}`) is replaced by the corresponding variable (e.g., `adjective`, `noun1`) that the user entered.

Why it's important:

- This step **builds the final story** by replacing the placeholders with actual words provided by the user, turning the template into a unique, personalized story.

6. **Display the Completed Story**

Explanation:

- This part simply prints the completed Mad Libs story to the console.
- We also print a separator (= * 50) to make the output more **visually organized** and readable.

Why it's important:

- **Displaying the story** is the main outcome of the program, and formatting it with a separator helps in making the output look cleaner and more engaging.

7. **Ask the User if They Want to Play Again**

Explanation:

- After showing the completed story, the program asks the user if they want to play again.

- **`.lower()`** ensures that the input is case-insensitive (so "YES", "Yes", or "yes" all work).
- If the user types "yes," the game restarts. If they type anything else (like "no"), the program exits.

Why it's important:

- This feature allows users to **replay** the game without restarting the program, enhancing usability and fun.
- It provides a way for the program to exit gracefully once the user is done playing.

8. **Main Program and Running the Game**

Explanation:

- The **`if __name__ == "__main__":`** block ensures that the `main()` function is only called when this file is executed directly, and not when it is imported as a module into another script.
- **`main()`** is the central function that runs the game. It controls the flow of generating stories and asking if the user wants to play again.

Why it's important:

- This ensures the program runs only when intended, making it easier to integrate into larger projects without unwanted execution when imported.