**Activities Module 6**

**Exercise 1: Concatenating with Variables**

Create your own example of concatenating strings using variables.

Choose two variables and concatenate them into a meaningful string.

**Exercise 2: Using Placeholder (f-strings)**

Write an example using f-strings to create a formatted string.

Use at least two variables to construct the formatted string.

**Exercise 3: Multi-line Concatenation**

Create an example that demonstrates multi-line concatenation.

Combine multiple lines of text into a single string for improved readability.

**4: Multi-line Concatenation and escape sequence**

Modify last exercise by adding escape sequence.

Escape sequences allow you to include special characters within strings and control the formatting of your text. They are particularly useful when you need to include characters that would otherwise have a special meaning in a string, such as quotes or newline characters.

\n - **Newline**: It represents a newline character, which moves the text to a new line, creating a line break.

\t - **Tab:** It represents a tab character, which is used for indentation. This can be useful for aligning text.

\\ - **Backslash Itself:** To include a single backslash in a string, you need to escape it with another backslash.

In the output, you'll see a single backslash.

\' - **Single Quotes:** To include a single quote within a single-quoted string, you can use the \' escape sequence.

\" - **Double Quotes:** To include a double quote within a double-quoted string, you can use the \" escape sequence.

**Exercise 5: PEP 8 Recommendations**

Review the provided PEP 8 recommendations.

Then, create a code snippet that adheres to these guidelines.

PEP 8 Recommendations:

1) Indentation: Use 4 spaces per indentation level.

2) Naming Conventions: Use snake\_case for variables and function names.

Use CamelCase for class names. Use UPPER\_CASE for constants.

3) Whitespace: Use spaces around operators and after commas.