8. Control Statements (Break, Continue, Pass)

❖Understanding the role of break, continue, and pass in Python loops.

In Python, loop control statements—break, continue, and pass—allow you to modify the flow of loops (for and while) to handle specific conditions more effectively. Here's an overview of each statement:

break: Exit the Loop Prematurely

The break statement terminates the loop entirely, regardless of whether the loop's condition is still true. This is useful when you want to stop looping based on a specific condition.

Syntax:

```
break
```

continue: Skip the Current Iteration

The continue statement skips the rest of the code inside the loop for the current iteration and moves to the next iteration. This is useful when you want to ignore certain conditions but continue looping.

Syntax:

```
continue
```

pass: Placeholder for Future Code

The pass statement is a null operation; it does nothing when executed. It's often used as a placeholder for future code or to create minimal classes or functions.

Syntax:

```
pass
```