1. Loop For: Repeat by number of times

While loop: Check condition and execute internal command, sometimes loop if not error.

2. break: Jumps out of the closest enclosing loop (past the entire loop statement)

continue: Jumps to the top of the closest enclosing loop (to the loop’s header line)

3. Runs if and only if the loop is exited normally (i.e., without hitting a break)

4. Counter loops can be coded with a while statement that keeps track of the index manually, or with a for loop that uses the range built-in function to generate successive integer offsets. Neither is the preferred way to work in Python, if you need to simply step across all the items in a sequence. Instead, use a simple for loop instead, without range or counters, whenever possible; it will be easier to code and usually quicker to run.

5. The range built-in can be used in a for to implement a fixed number of repetitions, to scan by offsets instead of items at offsets, to skip successive items as you go, and to change a list while stepping across it. None of these roles requires range, and most have alternatives—scanning actual items, three-limit slices, and list comprehensions are often better solutions today (despite the natural inclinations of ex–C programmers to want to count things!).