1. What are the main functional differences between a *while* and a *for*?

2. What’s the difference between *break* and *continue*?

3. When is a loop’s *else* clause executed?

4. How can you code a counter- based loop in Python?

5. What can a range be used for in a *for* loop?

1. *for* Loops allow us to run through the loop when we know how many times we’d like to run through the problem (Until any conditions is not met).

*while* Loops allow us to run through the loop when we actually don’t know exactly how many times we’d like to run throygh the problem, as the program will stop when a condition is met.

If we were to use an already existing variable in our program, then it’d be cleaner to just use a *while* loop. In the *for* Loops we must create a new variable.

2. We will *break* out of the loop once we’ve found what we’re looking for. Literally we’re done with this loop entirely.

We will *continue* with the remaining loops if we don’t need to keep going with the part of our loop that has to do with orders.

3. The loop *else* clause will be executed only if no previos factor is found. It also runs if the body of the loop is never executed, as we don’t run a break in that event.

4. We can code a counter-based loop in Python by either use a *while* statement or a *for* statement. The *while* statement keeps track of the index manually and the *for* loop uses the *range* function to generate successive integer offsets. However, if we need to simply step across all the items in a sequence, then instead we should use a simple *for* loop without *range* or counters as it will be easier to code and quicker to run.

5. A range can be used as a tool to implement a fixed number of repetitions or iterations, to scan by offsets instead of items at offsets, to skip successive items, and to change a list while stepping across it.