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

* The while loop is a general looping statement, but the for is designed to iterate

across items in a sequence or other iterable. Although the while can imitate the

for with counter loops, it takes more code and might run slower.

* The for statement iterates through a collection or iterable object or generator function.
* The while statement simply loops until a condition is False.
* It isn't preference. It's a question of what your data structures are.
* Often, we represent the values we want to process as a range (an actual list), or xrange (which generates the values). This gives us a data structure tailor-made for the for statement.
* Generally, however, we have a ready-made collection: a set, tuple, list, map or even a string is already an iterable collection, so we simply use a for loop.
* In a few cases, we might want some functional-programming processing done for us, in which case we can apply that transformation as part of iteration. The sorted and enumerate functions apply a transformation on an iterable that fits naturally with the for statement.
* If you don't have a tidy data structure to iterate through, or you don't have a generator function that drives your processing, you must use while.

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

* The break statement exits a loop immediately (you wind up below the entire while or for loop statement).
* The continue statement jumps back to the top of the loop (you wind up positioned just before the test in while or the next item fetch in for).

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

The else clause in a while or for loop will be run once as the loop is exiting, if the

loop exits normally (without running into a break statement).

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

Counter-based loops can be coded with:

* The while statement that keeps track of the index manually.
* The for loop that uses the range built-in function to generate successive integer offsets.
* The for statement is quicker to run than in the while statement.

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

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.