1/ What are the main functional differences between while and for?

* The while statement is the most general iteration construct in the language. It repeatedly executes a block of (normally indented) statements **as long as a test at the top keeps evaluating to a true value.**
* The for loop steps through the items **in any ordered sequence or other iterable object**. It works on strings, lists, tuples, and other built-in iterables, as well as new user-defined objects

2/ What is the difference between break and continue?

* The break statement causes an immediate exit from a loop
* The continue statement causes an immediate jump to the top of a loop

3/ When is a loop’s else clause executed?

* The else clause is executed when the loop is running normally without a break (break statement).

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

* Using while statement: you can put a variable for counting be assigning the value increasingly until the top of the loop has false value.
* Using for statement: you can use range built-in function to generate successive integer offsets. The for statement is quicker to run than in the while statement.

5/ What can a range be used in a for loop?

* Range is used most often to generate indexes in a for loop
* With one argument, range generates a list of integers from zero up to but not including the argument’s value.
* Range provides a simple way to repeat an action a specific number of times.
* The range call is also sometimes used to iterate over a sequence indirectly, though it’s often not the best approach in this role.
* For example, some algorithms can make use of sequence reordering—to generate alternatives in searches, to test the effect of different value orderings, and so on. Such cases may require offsets in order to pull sequences apart and put them back together, the range’s integers provide a repeat count in the first, and a position for slicing in the second.
* The range is for changing a list while stepping across it.