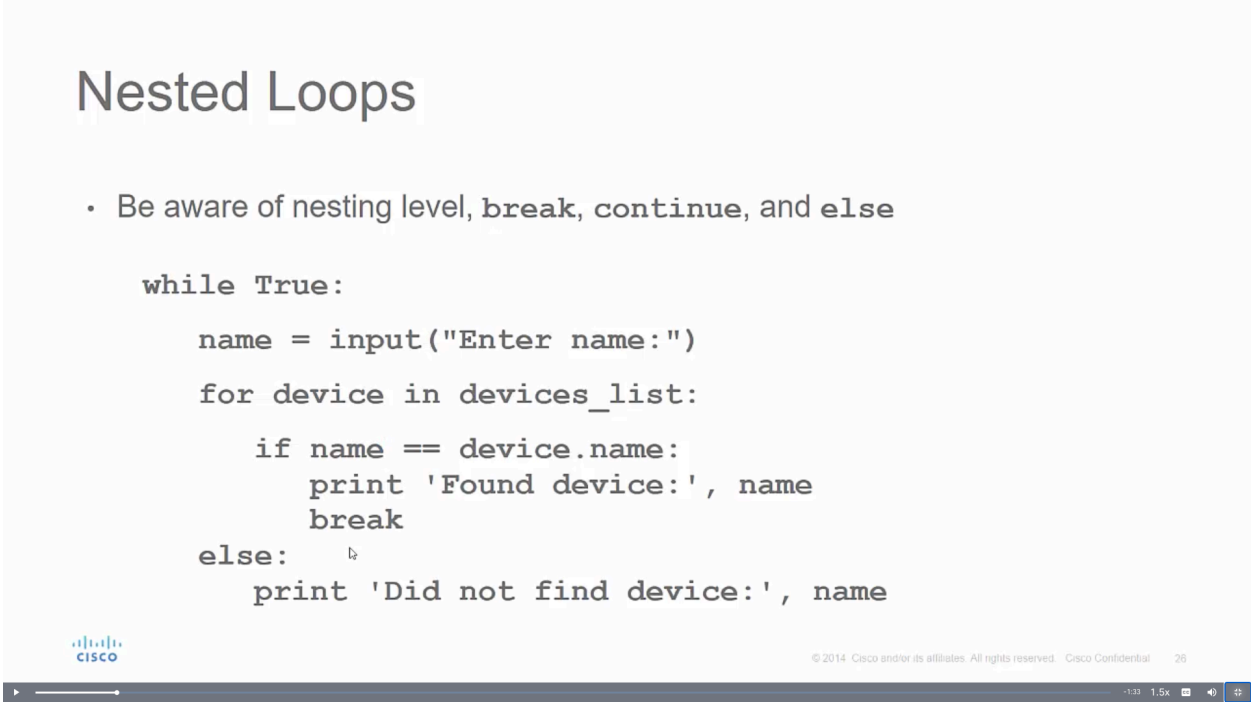

Sometimes it will be necessary to implement code that has multiple looping blocks nested within one another. In these situations, the inner loop will execute until it exits for some reason, continuing with the next iteration of the outer loop.

The following example shows an outer `while` loop, which continues to loop until some event causes the loop to exit (for example, user input). The details of exiting are not shown in this example for simplification purposes.

The inner loop iterates through all devices in a list of devices, looking for a specific device name. If the device name is found, the inner loop is exited, and iterations continue for the outer while statement.

```
while True:
   name = input("Enter name:")
   for device in devices_list:
      if name == device.name:
         print 'Found device:', name
         break
   else:
      print 'Did not find device:', name
```

Note: the inner `for` statement iterates over devices in the file; when a match is found, it is printed, and the `for` loop is exited. But look carefully at the `else` clause. It is important to understand where the `else` belongs. Looking closely at the indentation, you can see that the else belongs with the `for` statement.