When a `while` loop or a `for` loop is executed, the program will execute the body of the loop, circling and circling, as long as the conditions for the loop are met. In some scenarios, however, we might want to end the loop before it does, and Python provides two ways to force the body of the current loop to leave:

break

The `break` statement immediately terminates the execution of the current loop, exiting the loop structure in which it currently resides. Whenever a break statement is executed, either a `while` loop or a `for` loop, the body of the currently executing loop is stopped directly.

This is like running on the playground, originally planned to run 10 laps, but when the second lap, suddenly remembered something urgent to do, so decided to stop running and leave the playground, this is equivalent to using a break statement to pre-terminate the loop.

```
for i in range(11):
    if i == 5 :
        break
    print(i)
print("end")
```

The break statement is typically used in conjunction with an `if` statement to indicate that under certain conditions the body of the loop is broken.

continue

It simply terminates execution of the rest of the code in this loop and continues execution directly from the next loop.

Taking running as an example, I had planned to run 10 laps, but when I ran two and a half laps, I received a phone call and stopped running. When I hung up the phone, I started running the third lap instead of running the remaining half lap.

```
for i in range(11):
    print("before continue,", i)
    if i == 5 :
        continue
    print("after continue,", i)
print("end")
```