While Loop Statement

You can make a block of code execute over and over again using a `while` statement. The code in a `while` clause will be executed as long as the `while` statement's condition is `Ture`. In code, a while statement always consists of the following

- The `while` keyword
- A condition (that is, an expression that evaluates to `True` or `False`)
- A colon
- Starting on the next line, an indented block of code (called the `while` clause)

You can see that a `while` statement looks similar to an `if` statement. the difference is in how they behave. At the end of `if` clause, the program execution continues after the if statement, but at the end of `while`, clause, the program exection jumps back to the start of the `while` statement. The `while` clause is often called the `while loop` or just the `loop`.

example of while loop:

```
spam = 0
while spam < 5:
        print("Hello, world. ")
        spam = spam + 1
```

`while` check the value of `spam`, and if it's less than 5 they print a message. if it isn't then the print statement will again executing in lne 3.
On line 4 we have a handler to make sure that our code does not creating an infinited loop by increasing the value of spam so that condition on line 2 will no longer be `True`.

break statement

There is a shortcut to getting the program execution to break out of a `while` loop's earlier than turn the condition clause otherwise. if the execution reaches a `break` statement, it immediately exits the `while` loop's clause. In code, a `break` statement simply contains the `break` keyword.

below is a program to will allow user to break out of itself using `break` statement

```
while True:
        print("Please type your name: ")
        name = input()
        if name == 'your name':
                break
print('Thank you!')
```

continue statement

Like `break` statements, `continue` statements are used inside loops. When the program execution eachs a `continue` statement, the program exeuction immediately jump back to the start of the loop and re-evaluates the loop's condition. (This process also happens when the execution reached the end of the loop.)

```
while True:
        print("Who are you?")
        name = input()
        if name != 'Joe':
                continue
        print("Hello, Joe. What is the password?")

        password = input()
        if password == 'swordfish':
                break

print('Access granted.')
```

If the user enters any name beside Joe, the continue statement causes the program exection to jump back to the start of the loop. when the program re-evaluates the condition, the execution will always enter the loop, snce the condition is simply the `True`. Once the user makes it past that `if` statement, they are asked for a password. if the password entered is `swordfish`, then the `break` statement is run. and the execution jumps out of the `while` loop to print `Access granted.`. Otherwise, the exeuction continues to the end of the `while` loop, where it then jumps back to the start of the loop.