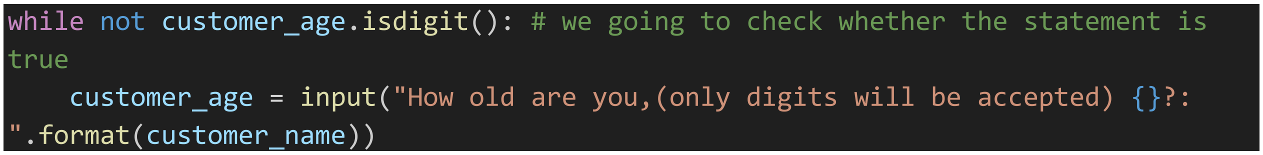
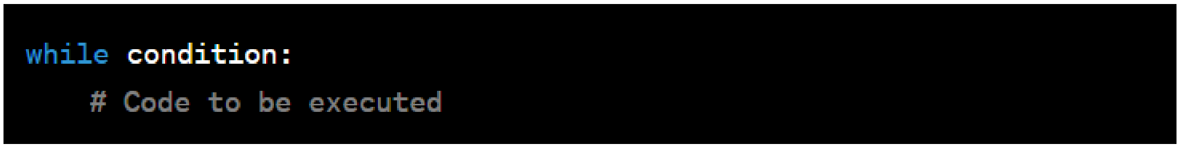
while loop:

- The **while** loop is used to repeatedly execute a block of code as long as **a given condition is true.**
- The loop first checks the condition. **If the condition is true, the code block inside the loop is** executed.
- Afterward, the condition is checked again, **and if it's still true,** the code block is executed again. This process continues until the condition becomes false.
- Here's the basic syntax of a **while** loop:



```
while not customer_age.isdigit(): # we going to check whether the statement is
true
    customer_age = input("How old are you,(only digits will be accepted) {}?:
".format(customer_name))
```

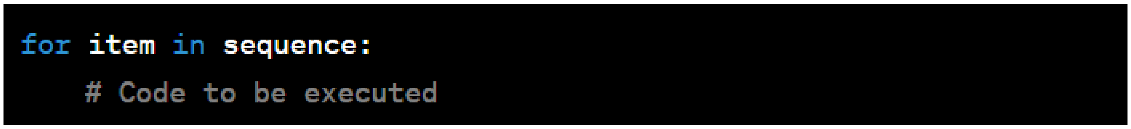
In the above code, the condition (not customer-age.isdigit()) will be evaluated. If it is **TRUE**, the block of code under it will be executed repeatedly until the condition is **FALSE**.



```
while condition:
    # Code to be executed
```

for loop:

- The **for** loop **is used to iterate over a sequence** (such as a list, tuple, or string) or other iterable objects.
- The **loop iterates over each item in the sequence, and for each iteration, the code block inside the loop is executed.**
- The **for** loop provides a convenient way to perform a specific action for each item in a collection.
- Here's the basic syntax of a **for** loop:



```
for item in sequence:
    # Code to be executed
```

The main difference between **while** and **for** loops is that **while** loops rely on a condition that is evaluated before each iteration, while **for** loops iterate over a predefined sequence.

In general, you can use a **while** loop when you don't know how many times the loop needs to be executed, and you can use a **for** loop when you have a specific sequence or collection of items to iterate over.