```
You entered:  20

Enter a number  :29

You entered:  29

Enter a number  :3

You entered:  3

Enter a number between :Traceback (most recent call last):

  File "test.py", line 5, in <module>

    num = raw_input("Enter a number :")

KeyboardInterrupt
```

Above example goes in an infinite loop and you need to use CTRL+C to exit the program.

Using else Statement with Loops

Python supports to have an **else** statement associated with a loop statement.

- If the **else** statement is used with a **for** loop, the **else** statement is executed when the loop has exhausted iterating the list.

- If the **else** statement is used with a **while** loop, the **else** statement is executed when the condition becomes false.

The following example illustrates the combination of an else statement with a while statement that prints a number as long as it is less than 5, otherwise else statement gets executed.

```
#!/usr/bin/python


count = 0

while count < 5:

    print count, " is  less than 5"

    count = count + 1

else:
```