Lesson 5 Python Loop Statement

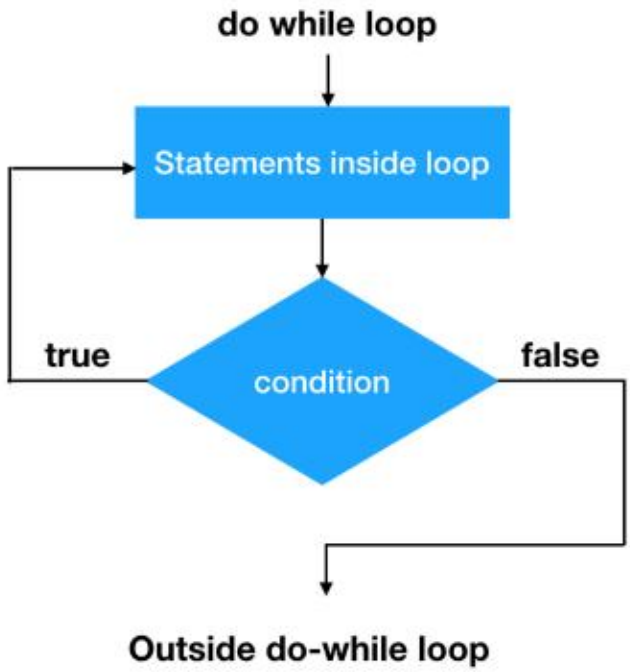
In the program, some steps of the algorithm will be executed repeatedly under certain condition, which is loop statement. In this lesson, Python statement will be explained combining with the related routines.

1. Loop Statement Introduction

Loop statement includes "**while**" and "**for**", which is used to repeat some steps.

1.1 while loop

while 循环的语法格式和执行流程如下：The format and execution process of while loop are as follow.

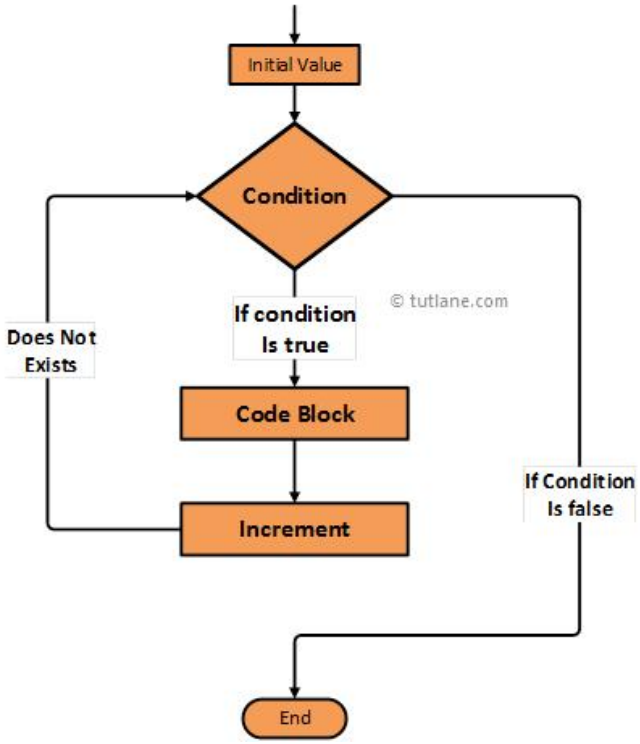
Syntax	Execution Process
`while loop condition:` `    code block`	

When the loop condition is "**True**", the block of code in while loop will be

executed till the loop condition is “**False**”. When the loop condition is always “**True**”, it will fall into “**dead loop**”.

1.2 for loop

The syntax and execution process of for loop are as follow.

Syntax	Execution Process
`for loop variable in target:` `    code block`	

”**Target**” can be string, list, tuple, dictionary and other sequence type, while “**loop variable**” is used to store the elements read from the variable of the sequence type

After entering **for loop**, traverse the elements within target and execute the block of code within **for loop** till the traversal ends.

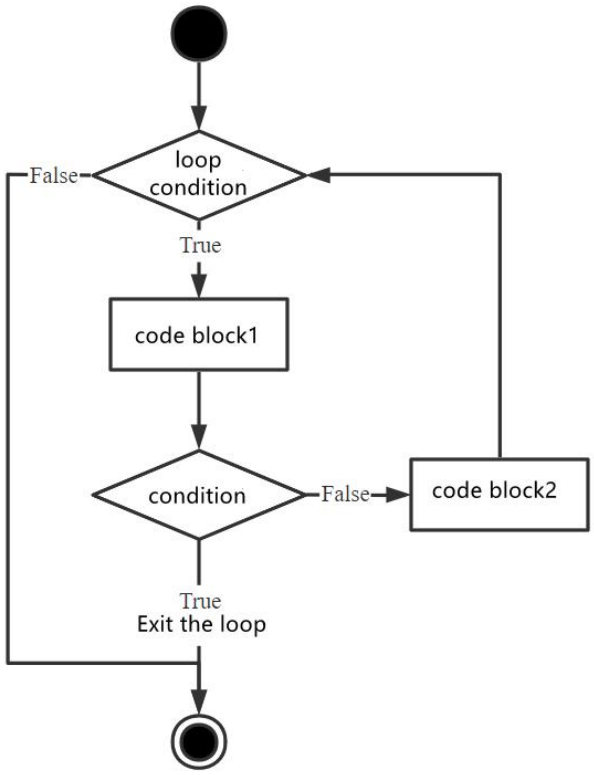
2. Loop Control Statement Introduction

Loop control statement can be used to interrupt the loop, or skip the current loop to execute the next loop. Loop control statement contains "**break**", "**continue**" and "**pass**".

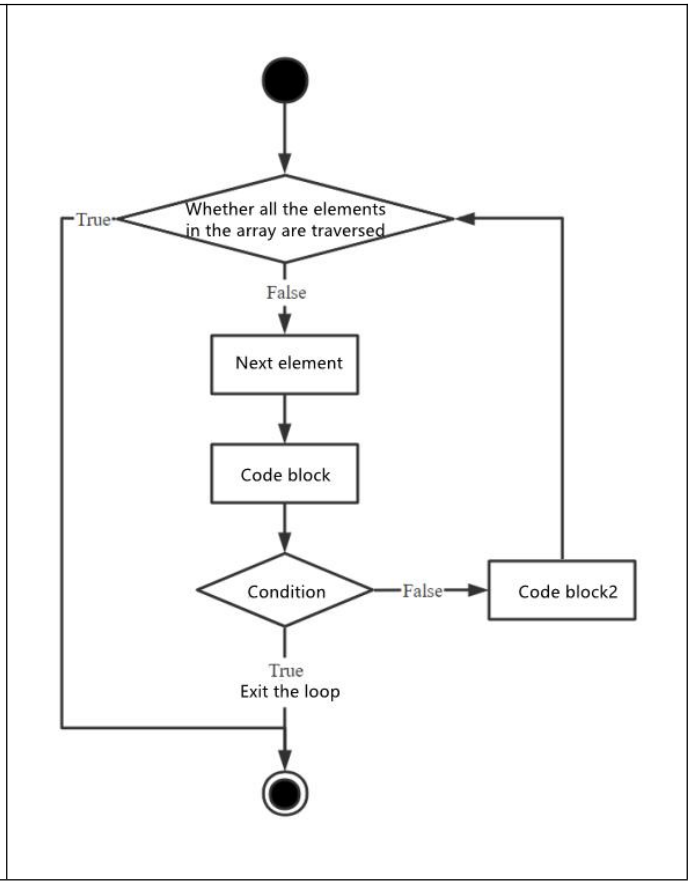
2.1 break Statement

break statement is used to get out of the whole loop. The syntax and execution process are as follow.

Syntax	Execution Process

`while loop condition:`
`  code block1`
`  if condition:`
`  break`
`  code block2`
	

```
for loop variable in target:
    code block1
    if condition:
        break
    code block2
```



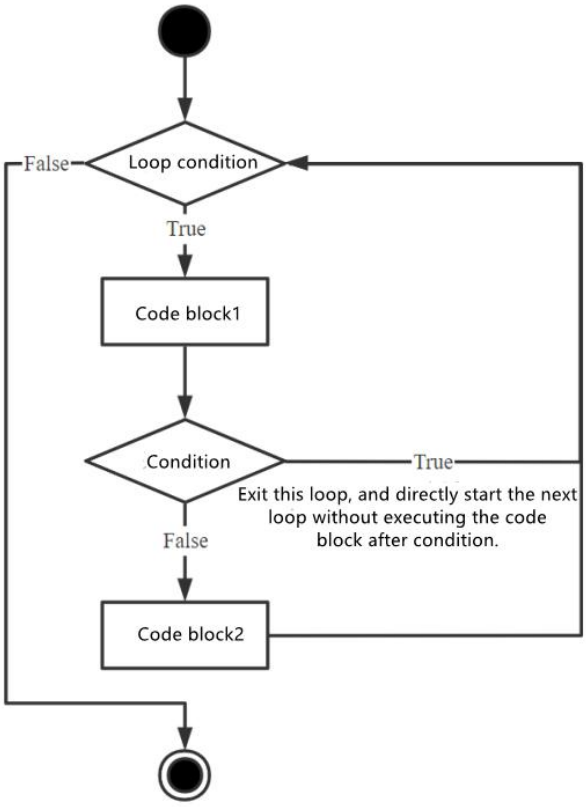
When the judgement condition is executed and is "**True**", the loop will end. If the judgement condition is "False", the block of code 2 will be executed and the loop will continue.

2.2 continue statement

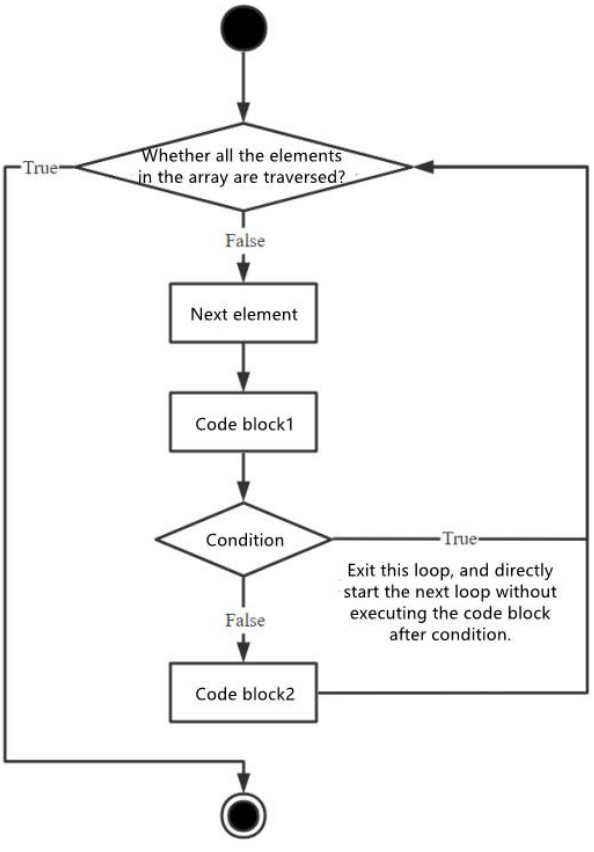
continue statement is used to get out of this round of loop and carry out the next round of loop. The syntax and execution process are as follow.

Syntax	Execution Process

```
while condition:
    code block1(0.001)
    if condition:
        continue
    code block2
```




```
for loop variable in target:
    code block1
    if condition:
        continue
    code block2
```



When the judgement statement is executed and is “**True**”, this round of loop is complete and next round of loop starts. If the judgement condition is “False”, the block of code 2 is executed and the second round of loop is done.

2.3 pass statement

pass statement is null statement and nothing will be executed. It is used to keep the integrity of the program structure.

3. Operation Steps

This routine will calculate the factorial of the integer and print the elements of the string.

Before operation, we need to copy the routine “**loop_statement.py**” stored in “**3. Python->Python Basic and Advanced Lesson-> Lesson 5 Python Loop Statement-> Routine Code**” to the shared folder.

For the configuration of the shared folder, please refer to the file in “**2. Linux Introduction and Usage->Linux Basic Lesson->Lesson 3 Linux Installation and Source Replacement**”.

Note: the input command should be case sensitive, and the keywords can be complemented by “Tab” key.

1） Start virtual machine, and click “  ”, and then click “  ” or press “**Ctrl+Alt+T**” to open command line terminal.

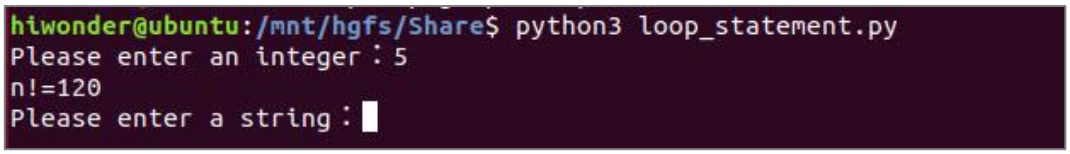
2） Input command “**cd /mnt/hgfs/Share/**” and press Enter to enter the shared folder.

```
hiwonder@ubuntu:~$ cd /mnt/hgfs/Share/
```

3） Input the command “**python3 loop_statement.py**” and press Enter to run the routine.

4. Program Outcome

Input one integer and press Enter, and then the terminal will print its factorial.



Input a string and press Enter, and then the terminal will print the elements of this string.

5. Program Analysis

The used routine “**loop_statement.py**” is stored in “**3. Python->Python Basic and Advanced Lesson-> Lesson 5 Python Loop Statement-> Routine Code**”.

```
n = int(input("请输入一个整数："))
fact = 1
i = 1
while i<= n:
    fact = fact*i
    i = i + 1
print("n!={}".format(fact))

string = input("请输入一个字符串：")
for c in string:
    print(c)
```

5.1 Calculate Factorial of the Value

1） Input data

Call input() function to receive the input data. The hints are inside the parenthesis.

```
n = int(input("请输入一个整数："))
```

2） Create variable

Create two variables for later calculation. “**fact**” is used to store the current factorial.

```
fact = 1
i = 1
```

3） while loop

Use while loop statement to calculate the factorial of the designated integer. During loop, “**i**” value will keep adding up. When this value is greater than the input integer, the loop will end. Then, call print() function to print the result on the terminal.

```
while i<= n:
    fact = fact*i
    i = i + 1
print("n!={}".format(fact))
```

The syntax of print() function is as follow.

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

The first parameter "**objects**" is the output object. When output several objects for one time, objects should be separated by "**,**".

The second parameter "**sep**" is used to insert the string between the objects. The default value is a space.

The third parameter "**end**" is used to add string at the end of the output. The default value is a line break.

The fourth parameter "**file**" is the object with a write function, the default value is "**sys.stdout**", that is screen.

The fifth parameter "**flush**" is used to control and output cache. The default value is "**False**".

5.2 Print String Elements

1) Input data

Call input() function to receive the input data. The hints are inside the parenthesis.

```
9  string = input("请输入一个字符串：")
```

2) For loop

Use for loop to obtain the elements of the input string. Then call print() function in the loop structure to print the input string separately.

```
10 for c in string:
11     print(c)
```