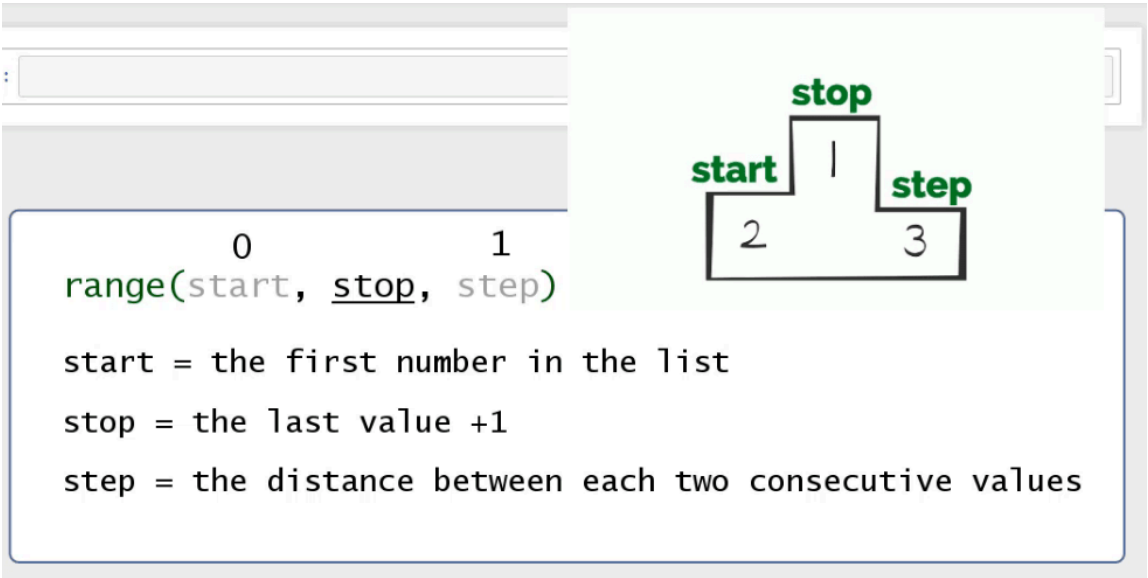
Loops in Python

Table of Contents

Range

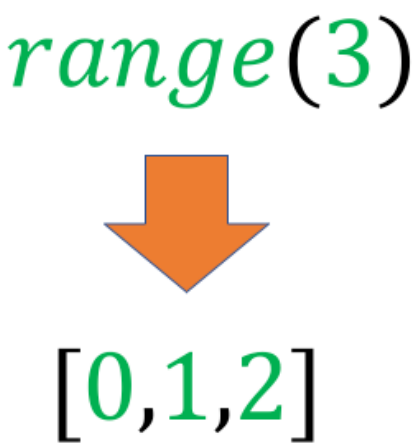
Sometimes, you might want to repeat a given operation many times. Repeated executions like this are performed by **loops**. We will look at two types of loops, `for` loops and `while` loops.

Before we discuss loops lets discuss the `range` object. It is helpful to think of the range object as an ordered list. For now, let's look at the simplest case. If we would like to generate a sequence that contains three elements ordered from 0 to 2 we simply use the following command:



In [1]:
```
# Use the range
print(range(3))   # (0,3,1)
```

```
range(0, 3)
```



Loops

What is `for` loop?

The `for` loop enables you to execute a code block multiple times.

In this example we can print out a sequence of numbers from 0 to 7:

In [2]:
```
# range(3)
for i in range(0,3,1):
    print(i)
```

```
0
1
2
```

In [3]:
```
t = 2
for i in range(11):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
```

In [6]:
```
t = 2
for i in range(10):
    print(t, "x", i+1, "=", (i+1)*t)
```

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

In [7]:
```
t = int(input("Enter the Multiplication Table: "))
for i in range(1, 11):
    print(t, "x", i, "=", i*t)
```

```
Enter the Multiplication Table: 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
```

In [8]:
```
Lst = [5,3,6,8,9,2,7]
```

In [9]:
```
for i in Lst:
    print(i)
```

```
5
3
6
8
9
2
7
```

In [10]:
```
Tab = [3,5,8]
```

In [17]:
```
for t in Tab:
    print(f"====={t}=====")
    for i in range(1, 6):
        print(t, "x", i, "=", t * i)
```

```
====3=====
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
====5=====
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
====8=====
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
```

In [18]:
```python
Tab = []
for i in range(3):
    T = int(input(f"Enter Table {i+1}: "))
    Tab.append(T)

print(Tab)
```

```
Enter Table 1: 4
Enter Table 2: 9
Enter Table 3: 13
[4, 9, 13]
```

In [19]:
```python
for t in Tab:
    print(f"====={t}=====")
    for i in range(1, 6):
        print(t, "x", i, "=", t * i)
```

```
====4=====
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
====9=====
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
====13=====
13 x 1 = 13
13 x 2 = 26
13 x 3 = 39
13 x 4 = 52
13 x 5 = 65
```

In [23]:
```python
for i in range(0,4,1):
    print("* " * (i+1))
```

```
*
* *
* * *
* * * *
```

In [30]:
```python
for i in range(4,0,-1):
    print("* " * i)
```

```
* * * *
* * *
* *
*
```

In [50]:
```
Lst = [23, 66, 41, 9, 78, 33]
for i in Lst:
    if i % 2 == 0:
        print(i)
```

```
66
78
```

In [51]:
```
for i in Lst:
    if i % 2 == 0:
        print(i, "Even")
    else:
        print(i, "Odd")
```

```
23 Odd
66 Even
41 Odd
9 Odd
78 Even
33 Odd
```

In [54]:
```
eLst = []
oLst = []
for i in Lst:
    if i % 2 == 0:
        eLst.append(i)
    else:
        oLst.append(i)

print("Even Numbers List: ", eLst)
print(" Odd Numbers List: ", oLst)
```

```
Even Numbers List:  [66, 78]
 Odd Numbers List:  [23, 41, 9, 33]
```

In [35]:
```
for i in Lst:
    print(i)
```

```
23
66
41
9
78
33
```

In [36]:
```
sm = 0
for i in Lst:
    sm +=i

print(sm)
```

```
250
```

In [41]:
```
mn = Lst[0]
for i in Lst:
    if i < mn:
        mn = i

print(mn)
```

```
9
```

In [42]:
```
mx = Lst[0]
for i in Lst:
    if i > mx:
        mx = i

print(mx)
```

```
78
```

In [45]:
```
ct = 0
for i in Lst:
    ct += 1

print(ct)
```

```
6
```

In [46]:
```
avg = sm / ct
avg
```

Out[46]:
```
41.666666666666664
```

In [18]:
```
sm, cnt = 0, 0    #  sm = 0; cnt =0
for i in Lst:
    sm = sm + i
    cnt = cnt + 1

print(sm / cnt)
```

```
41.666666666666664
```

In [19]:
```
Lst
```

Out[19]:
```
[23, 66, 41, 9, 78, 33]
```

In [8]:
```
a = int(input("Enter a Number: "))
res = "Not Found"
for i in Lst:
    #print(i)
    if i == a:
        res = "Found"
        break
print(a, res)
```

```
Enter a Number: 78
78 Found
```

In [49]:
```
i ='python'
print(i[::-1])
```

```
nohtyp
```

In [47]:
```
txt =['python','civic','mongodb', 'radar']
```

In [48]:
```
for i in txt:
    if i == i[::-1]:
        print(i, " is a Plaindrome")
    else:
        print(i, " is NOT a Plaindrome")
```

```
python  is NOT a Plaindrome
civic  is a Plaindrome
mongodb  is NOT a Plaindrome
radar  is a Plaindrome
```

In [ ]:

```
```

In [11]:

```
for t in range(2,4,1):      # 2,3
    for m in range(1,6,1):  # 1,2,3,4,5
        print(t, "x", m, "=", t * m)
    print("----------")
```

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
----------
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
----------
```

In [2]:

```
for m in range(1,6,1):
    for t in range(2,4,1):
        print(t, "x", m, "=", t * m, end="\t")
    print("")
```

```
2 x 1 = 2	3 x 1 = 3
2 x 2 = 4	3 x 2 = 6
2 x 3 = 6	3 x 3 = 9
2 x 4 = 8	3 x 4 = 12
2 x 5 = 10	3 x 5 = 15
```

In [22]:

```
for m in range(1,6,1):
    for t in [3,8,13]:
        print(t, "x", m, "=", t * m, end="\t")
    print("")
```

```
3 x 1 = 3	8 x 1 = 8	13 x 1 = 13
3 x 2 = 6	8 x 2 = 16	13 x 2 = 26
3 x 3 = 9	8 x 3 = 24	13 x 3 = 39
3 x 4 = 12	8 x 4 = 32	13 x 4 = 52
3 x 5 = 15	8 x 5 = 40	13 x 5 = 65
```

In [3]:

```
for i in range(10):
    for j in range(10-i):
        print("-", end="")
    for k in range(i+1):
        print("*", end="")
    print()
```

```
----------*
---------**
--------***
-------****
------*****
-----******
----*******
---********
--*********
-**********
```

In [10]:

```
for i in range(10):
    for j in range(10-i):
        print("*", end="")
#    for k in range(i+1):
```

```
#         print("-", end="")
    print()
```

```
**********
*********
********
*******
******
*****
****
***
**
*
```

In [7]:
```
for i in range(10):
    for j in range(i+1):
        print("-", end="")
    for k in range(10-i):
        print("*", end="")
    print()
```

```
-**********
--*********
---********
----*******
-----******
------*****
-------****
--------***
---------**
----------*
```

In [9]:
```
for i in range(10):
    for j in range(i+1):
        print("*", end="")
#    for k in range(10-i):
#        print("-", end="")
    print()
```

```
*
**
***
****
*****
******
*******
********
*********
**********
```

What is `while` loop?

As you can see, the `for` loop is used for a controlled flow of repetition. However, what if we don't know when we want to stop the loop? What if we want to keep executing a code block until a certain condition is met? The `while` loop exists as a tool for repeated execution based on a condition. The code block will keep being executed until the given logical condition returns a **False** boolean value.

Let's say we would like to iterate through list `dates` and stop at the year 1973, then print out the number of iterations. This can be done with the following block of code:

In [7]:
```
# While Loop Example

dates = [1982, 1980, 1973, 2000]

i = 0
year = 0

while(year != 1973):
    year = dates[i]
    i = i + 1
    print(year)

print("It took ", i ,"repetitions to get out of loop.")
```

```
1982
1980
1973
It took  3 repetitions to get out of loop.
```

A while loop iterates merely until the condition in the argument is not met.

Write a while loop to display the values of the Rating of an album playlist stored in the list `PlayListRatings`. If the score is less than 6, exit the loop. The list `PlayListRatings` is given by: `PlayListRatings = [10, 9.5, 10, 8, 7.5, 5, 10, 10]`

In [12]:
```
# Write your code below and execute
```

In [13]:
```
Lst = [10, 9.5, 10, 8, 7.5, 5, 10, 10]
i = 1
Rating = Lst[0]
while(Rating >= 6):
    print(Rating)
    Rating = Lst[i]
    i = i + 1
```

```
10
9.5
10
8
7.5
```

Write a while loop to copy the strings `'orange'` of the list `squares` to the list `new_squares`. Stop and exit the loop if the value on the list is not `'orange'`:

In [15]:
```
colors = ['orange', 'orange', 'purple', 'blue ', 'orange']
new_colors = []
i = 0

while(colors[i] == 'orange'):
    new_squares.append(colors[i])
    i = i + 1
print (new_colors)
```

```
['orange', 'orange']
```

In [ ]: