For loop

execute a set of statements, once for each item in a list

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

```
apple
banana
cherry
```

```
for x in "banana":
  print(x)
```

What will be the output?

```
b
a
n
a
n
a
```

The break Statement

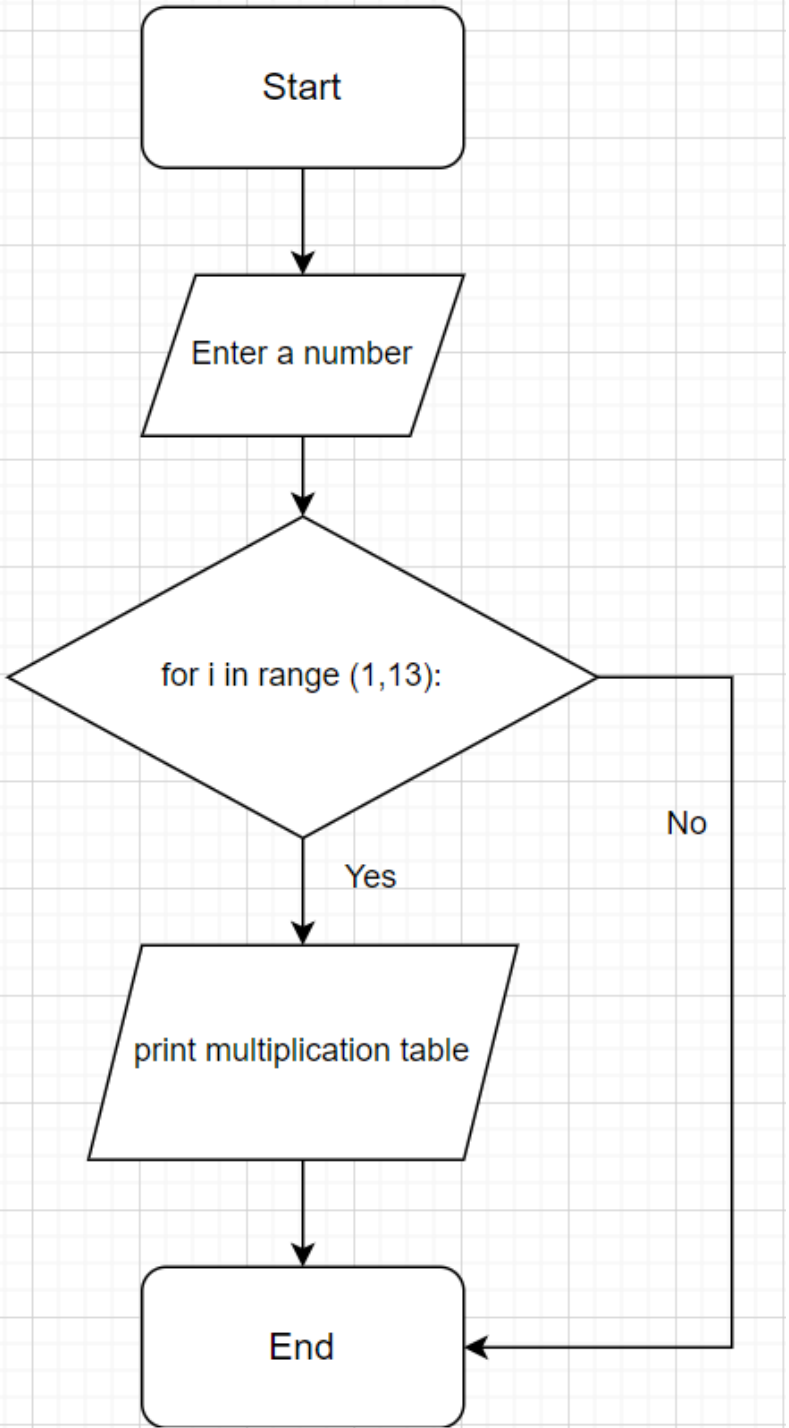
With the break statement we can stop the loop before it has looped through all the items:

Example

Exit the loop when `x` is "banana":

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

Question 1



Ask the user to enter any integer number

Display the multiplication table (between 1 to 12) for that number.

Ask the user to enter any integer number

Display the multiplication table (between 1 to 12) for that number.

Use only for-loop for the iterative process.

```
= RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Py
Enter a number to display the multiplication table: 567
1  x 567 =  567
2  x 567 = 1134
3  x 567 = 1701
4  x 567 = 2268
5  x 567 = 2835
6  x 567 = 3402
7  x 567 = 3969
8  x 567 = 4536
9  x 567 = 5103
10 x 567 = 5670
11 x 567 = 6237
12 x 567 = 6804
```

```
user_input = int(input('Enter a number to display the multiplication table: '))

for i in range (1,13):
    print (f'{i: <2} x {user_input: >2} = {i*user_input:>3}')
```

10 x 567 = 5670

Right side – 3 spaces

```
Enter any integer number: 45
Multiplication table for 45:
45 x 1 = 45
45 x 2 = 90
45 x 3 = 135
45 x 4 = 180
45 x 5 = 225
45 x 6 = 270
45 x 7 = 315
45 x 8 = 360
45 x 9 = 405
45 x 10 = 450
45 x 11 = 495
45 x 12 = 540
```

Not beautiful

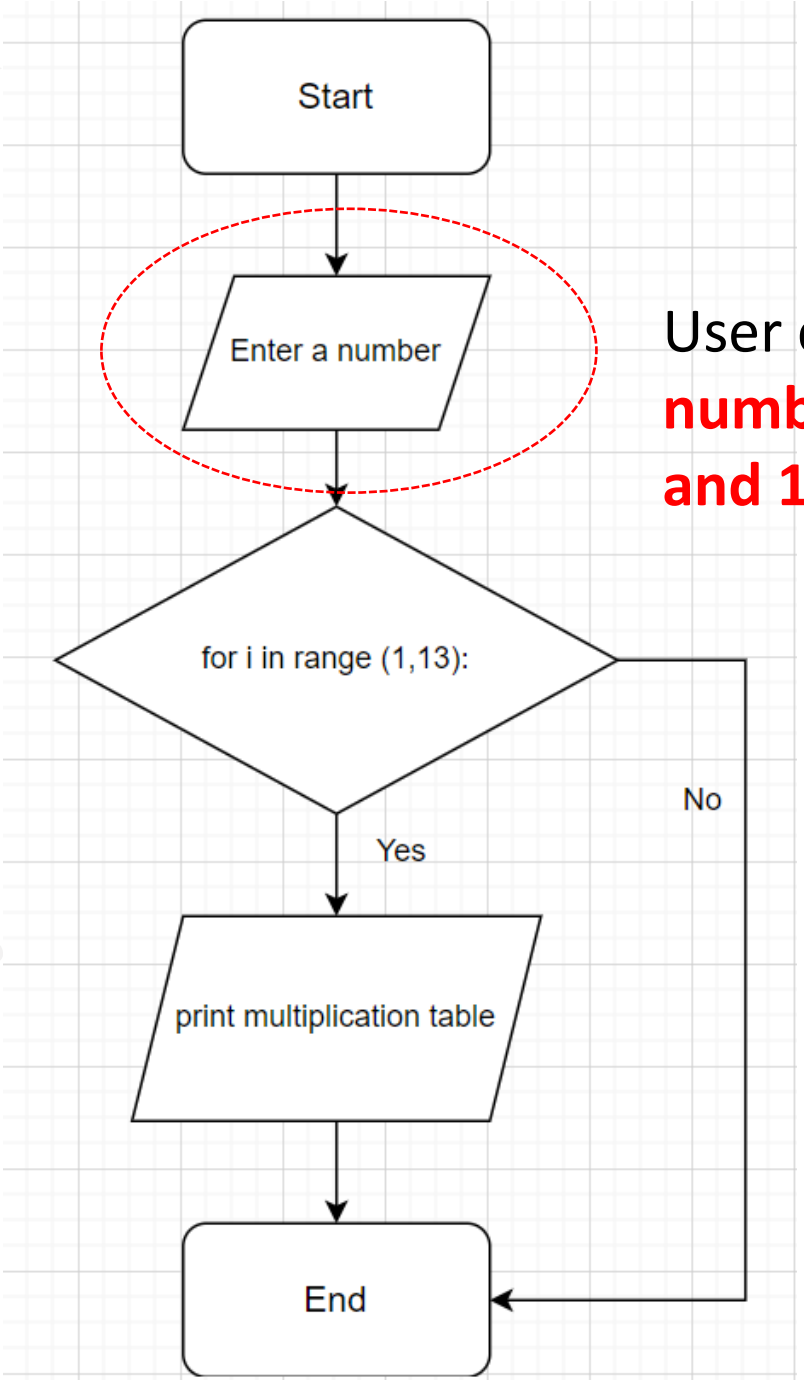
```python
# Ask the user to enter any integer number
number = int(input("Enter any integer number: "))

# Display the multiplication table for the entered number
print(f"Multiplication table for {number}:")
for i in range(1, 13):
    print(f"{number} x {i} = {number * i}")
```

Another way to write the code

Ask the user to enter a number between 1 and 12 and

then **display the multiplication table for that number.**

Use only for-loop for the iterative process.



User can only enter a **number between 1 and 12**

What will you modify?

```python
# Ask the user to enter a number between 1 and 12
number = int(input("Enter a number between 1 and 12: "))

# Validate user input
if number < 1 or number > 12:
    print("Invalid input")
else:
    # Display the multiplication table for the entered number
    print(f"Multiplication table for {number}:")
    for i in range(1, 13):
        print(f"{number} x {i} = {number * i}")
```

```
= RESTART: C:/Users/warhlaingn/AppDa
py
Enter a number between 1 and 12: 45
Invalid input
```

```
= RESTART: C:/Users/warhlaingn/AppData
py
Enter a number between 1 and 12: 12
Multiplication table for 12:
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
12 x 10 = 120
12 x 11 = 132
12 x 12 = 144
```

Question 2

Ask for a number below 50

Count down from 50 to that number

Make sure you show the number entered in the output

Use only for-loop for the iterative process

```
>>>
= RESTART: C:/Users/warhlaingn/
Enter a number below 50: 50
Invalid input
>>>
= RESTART: C:/Users/warhlaingn/
Enter a number below 50: 49
Counting down from 50 to 49:
50
49
>>>
= RESTART: C:/Users/warhlaingn/
Enter a number below 50: 39
Counting down from 50 to 39:
50
49
48
47
46
45
44
43
42
41
40
39
```

Start

Input: Ask for a number below 50

Check if the entered input is below 50

|

No ----> Display an error message and End

|

Yes

|

Display: "Counting down from 50 to entered number"

For each number **from 50 down to the entered number:**

|

|----> Display the current number

|

End

Flowchart???

```
# Ask the user to enter a number below 50
number = int(input("Enter a number below 50: "))

# Validate user input
if number >= 50:
    print("Invalid input")
else:
    # Display the countdown from 50 to the entered number
    print(f"Counting down from 50 to {number}:")
    for i in range(50, number - 1, -1):
        print(i)
```

OR

```
try:
    while True:
        user_input = int(input("Enter a number less than or equal to 50: "))
        if user_input <= 50:
            break
except:
    print('Please enter a valid input!!')
else:
    for i in range (50, user_input -1, -1):
            print (f'{i:<2}')
```

Question 3

Ask which direction the user wants to count (count up or count down).

If they select **up**, then ask them for the top number and count from 1 to that number.

If they select **down**, ask them to enter a number below 20 and then count down from 20 to that number.

If they entered a selection for something other than up or down, display an error message "I don't understand".

Use only for-loop for the iterative process.

```
>>>
==== RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Python/Pyt
Do you want to count up or count down? (Type 'up' or 'down'): up
Enter the top number: 5
Counting up:
1
2
3
4
5
>>>
==== RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Python/Pyt
Do you want to count up or count down? (Type 'up' or 'down'): down
Enter a number below 20: 5
Counting down from 20:
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
```

```
>>>
==== RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Python/Python3
Do you want to count up or count down? (Type 'up' or 'down'): UPDOWN
I don't understand. Please select 'up' or 'down'.
>>>
==== RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Python/Python3
Do you want to count up or count down? (Type 'up' or 'down'): down
Enter a number below 20: 23
Invalid input. Please enter a number below 20.
>>>
```

Start

Ask the user for the counting direction (up or down)

|

|--> If direction is up

| |

| Ask for the top number

| |

| Display: "Counting up"

| For each number from 1 to top number

| |

| Display the current number

| End

|

|--> If direction is down

| |

| Ask for a number below 20

| |

| If the entered number is not below 20

| | |

| | Display an error message and End

| |

| Display: "Counting down from 20"

| For each number from 20 down to the entered number

| |

| Display the current number

| End

|

|--> If direction is neither up nor down

| |

| Display an error message and End

|

End

Flowchart???

```
# Ask the user for the direction to count
direction = input("Do you want to count up or count down? (Type 'up' or 'down'):

# Check the direction and proceed accordingly
if direction == 'up':
    # If counting up, ask for the top number
    top_number = int(input("Enter the top number: "))

    # Display counting up from 1 to the top number
    print("Counting up:")
    for i in range(1, top_number + 1):
        print(i)
elif direction == 'down':
    # If counting down, ask for a number below 20
    number_below_20 = int(input("Enter a number below 20: "))

    # Validate the entered number
    if number_below_20 >= 20:
        print("Invalid input")
    else:
        # Display counting down from 20 to the entered number
        print("Counting down from 20:")
        for i in range(20, number_below_20 - 1, -1):
            print(i)
else:
    # If the direction is neither 'up' nor 'down', display an error message
    print("I don't understand")
```

What is an Exception?

In Python, **an exception is an error object.** It is an error that occurs during the execution of your program and stops it from running – subsequently displaying an error message.

When an exception occurs, Python creates an exception object which contains the type of the error and the line it affects.

try...except Syntax

Instead of allowing these exceptions to stop your program from running, you can put the code you want to run in a try block and handle the exception in the except block.

The basic syntax of try...except looks like this:

```
try:
    # code to run
except:
    # handle error
```

try...except Syntax

For example, if you have a large program and you don't know whether an identifier exists or not, you can execute what you want to do with the identifier in a try block and handle a possible error in the except block:

```
try:
  print("Here's variable x:", x)
except:
  print("An error occured") # An error occured
```

match case Statement

match case statements looks a lot like an if statement in Python.

```
>>> command = 'Hello, World!'
>>> match command:
...     case 'Hello, World!':
...         print('Hello to you too!'
...     case 'Goodbye, World!':
...         print('See you later')
...     case other:
...         print('No match found')

Hello to you too!
```

```
parameter = "Geeksforgeeks"

match parameter:

    case first  :
        do_something(first)

    case second :
          do_something(second)

    case third :
        do_something(third)
        ............
        ............
    case n :
        do_something(n)
    case _  :
          nothing_matched_function()
```

```
try:
    print('Do you want to count up or down?')
    print('[1] Count Up')
    print('[2] Count Down')
    selection = int(input('Enter your selection: '))

    match selection:
        case 1:
            num = int (input ('What is the top number? '))

            for i in range (1, num + 1):
                print (i)

        case 2:
            while True:
                num = int(input ('Enter a number below 20: '))

                if num < 20:
                        break

                for i in range(20, num -1, -1):
                        print (i)

        case _:
            print('I don\'t understand')

except:
    print('Enter a valid value')
```

Question 4

Make a quiz that asks five questions by randomly generating two whole numbers to make the question (for example: num1 + num2).

Ask the user to enter the answer.

If they get it right add a point to their score.

At the end of the quiz, tell them how many they got correct out of five. Use only for-loop for the iterative process.

```
= RESTART: C:/Users/warhlaingn
75 + 52 =
Your answer: 100

84 + 71 =
Your answer: 50

13 + 61 =
Your answer: 74

22 + 90 =
Your answer: 112

4 + 4 =
Your answer: 8

Your scored 1 out of 5
```

Flowchart

Start

Initialize score to 0

For each question in range(1, 6):
  |
  |--> Generate two random whole numbers (num1 and num2)
  |--> Calculate the correct answer (correct = num1 + num2)
  |--> Display the question: "num1 + num2 = ?"
  |--> Ask the user to enter the answer
  |--> Check if the user's answer is correct
  | |
  | |--> If correct:
  | | |--> Increment score by 1
  | |
  | |--> If incorrect:
  | | |--> Do nothing
  |
Display the final score: "You scored [score] out of 5."

End

```
import random
score = 0

for i in range (1,6):
    num1 = random. randint (1, 100)
    num2 = random. randint (1, 100)
    correct = num1 +num2
    print (num1, '+', num2, '=')

    answer = int(input('Your answer: '))
    print ()

if answer == correct:
    score += 1
print ('Your scored', score, 'out of 5')
```

Question 5

Redesign the Python program of Question 4 Coding Exercise 4 to use for-loop for all iterative processes, except the main loop that prompts the user to repeat the program can maintain using the while-loop.

4. Write a temperature conversion program between degree Celsius and degree Fahrenheit, according to user selection. The following is the formula:

```
Celsius = (Fahrenheit - 32) * 5/9

Fahrenheit = (Celsius * 9/5) + 32
```

- The temperature conversion program is expected to convert a range of temperature measurements from a minimum value to a maximum value.
- The program will only execute if the minimum temperature is smaller than or equal to the maximum temperature. Else, an error message should be prompted to the user and the program restart.
- The program will only execute when the **menu selection is valid**. Else, an error message should be prompted to the user and the program restart.
- **Write an iteration check if the user would like to run the program again.**

```
play_again = 'Y'

while play_again == 'Y':

    print ('Temperature Conversion Programme.')
    print ('[1] Convert Celsius to Fahrenheit.')
    print ('[2] Convert Fahrenheit to Celsius.')

    selection = int (input('Enter your selection, 1 or 2: '))

    if selection == 1:
        print ('Celsius (C) to Fahrenheit (F) Conversion')
        print ('Enter temperature in interger values only.')
        temp_min = int(input('Enter minimum temperature: '))
        temp_max = int(input('Enter maximum temperature: '))

        if temp_min <= temp_max:
            temp_c = temp_min
            while temp_c <= temp_max:
                temp_f = (temp_c * 9/5) + 32
                print(f'{temp_c:>5.1f}C = {temp_f:5.1f}F')
                temp_c = temp_c + 1
            print ('Conversion Done.')
        else:
            print ('Error: Invalid Input!')
```

```
    elif selection == 2:
        print ('Fahrenheit (F) to Celsius (C) Conversion')
        print ('Enter temperature in interger values only.')
        temp_min = int(input('Enter minimum temperature: '))
        temp_max = int(input('Enter maximum temperature: '))

        if temp_min <= temp_max:
            temp_f = temp_min
            while temp_f <= temp_max:
                temp_c = (temp_f-32)*5/9
                print(f'{temp_f:>5.1f}F = {temp_c:5.1f}C')
                temp_f = temp_f+1
            print ('Conversion Done.')
        else:
            print ('Error: Invalid Input!')
    else:
        print ('Error: Invalid Selection!')

    play_again = input('Do you want to run the program again? [Y/N]: ').upper()
    while play_again != 'N' and play_again != 'Y':
       play_again = input('Do you want to run the program again? [Y/N]: ').upper()
print ('Program Terminated.')
```

What will be the new code?

Try it out by yourself!!!

Question 5

Write a program to read ten (10) numbers into a list.

Then, traverse the list using a for-loop to find the maximum and minimum numbers using the ">" and “<" operators.

The program should produce the following outputs.

```
A program to find the maximum and minimum numbers in a list.
Enter ten (10) numbers into a list.
Enter a number: 10
Enter a number: -10
Enter a number: 30
Enter a number: 40
Enter a number: 99.99
Enter a number: 33
Enter a number: 78
Enter a number: 45
Enter a number: 35
Enter a number: 22
my_list = [10.0, -10.0, 30.0, 40.0, 99.99, 33.0, 78.0, 45.0, 35.0,
22.0]
Maximum Number = 99.99
Minimum Number = -10.0
```

Flowchart

| Start |

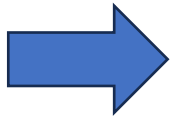
| my_list |

| Print Message |

| Input Numbers |

| Append Numbers |

| to my_list |



| Initialize |

| num_max, |

| num_min |

| Find Maximum |

| and Minimum |

| Numbers |

| Print Results |

| End |

Write a program to read ten (10) numbers into a list.

Then, traverse the list using a for-loop to find the maximum and minimum numbers using the ">" and “<" operators.

```
my_list = []

print ("A program to find the maximum and minimum numbers in a list.")
print ("Enter ten (10) numbers into a list.")

for i in range (1,11):
    my_num = float(input("Enter a number: "))
    my_list.append (my_num)

num_max, num_min = my_list [0], my_list [0]

for num_list in my_list:
    if num_list > num_max:
        num_max = num_list
    if num_list < num_min:
        num_min = num_list

print ("my_list = ", my_list)
print ("Maximum number = ", num_max)
print ("Minimum number = ", num_min)
```

```
= RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Python/Pythor
A program to find the maximum and minimum numbers in a list.
Enter ten (10) numbers into a list.
Enter a number: 1
Enter a number: 2
Enter a number: 3
Enter a number: 5
Enter a number: 7
Enter a number: 9
Enter a number: 10
Enter a number: 12
Enter a number: 15
Enter a number: 17
my_list =  [1.0, 2.0, 3.0, 5.0, 7.0, 9.0, 10.0, 12.0, 15.0, 17.0]
Maximum number =  17.0
Minimum number =  1.0
```