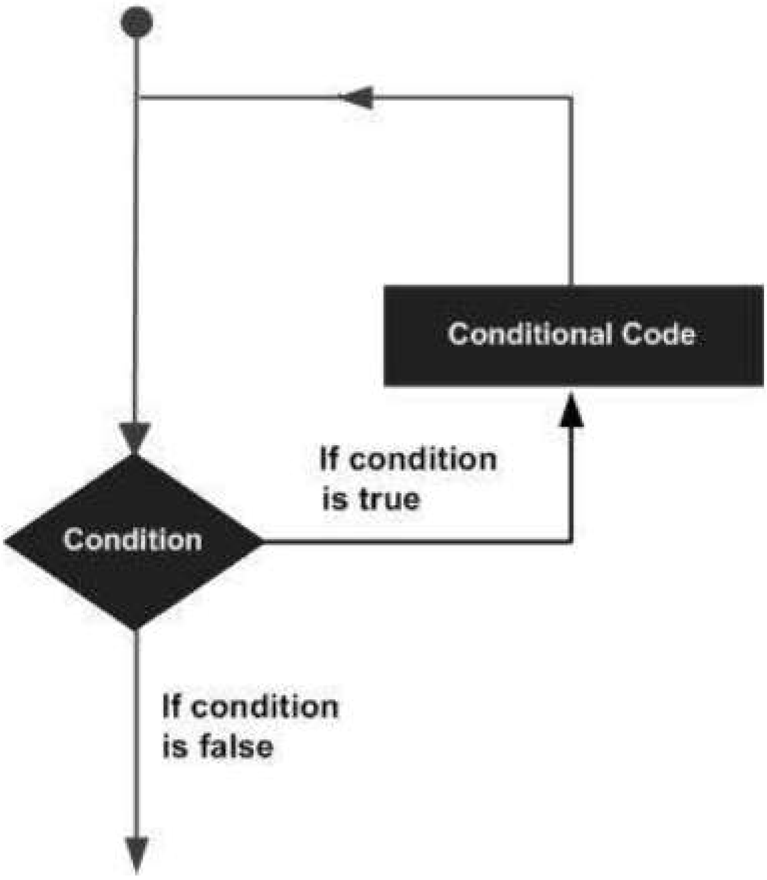
Agenda:

- While loop in Python

Basics:

- In general, statements are executed sequentially.
  - The first statement in a function is executed first, followed by the second, and so on.
  - There may be a situation when you need to execute a block of code several number of times.
- Repeats a statement or group of statements while a given condition is TRUE.
- It tests the condition before executing the loop body.
- A loop statement allows us to execute a statement or group of statements multiple times. The following diagram illustrates a loop statement

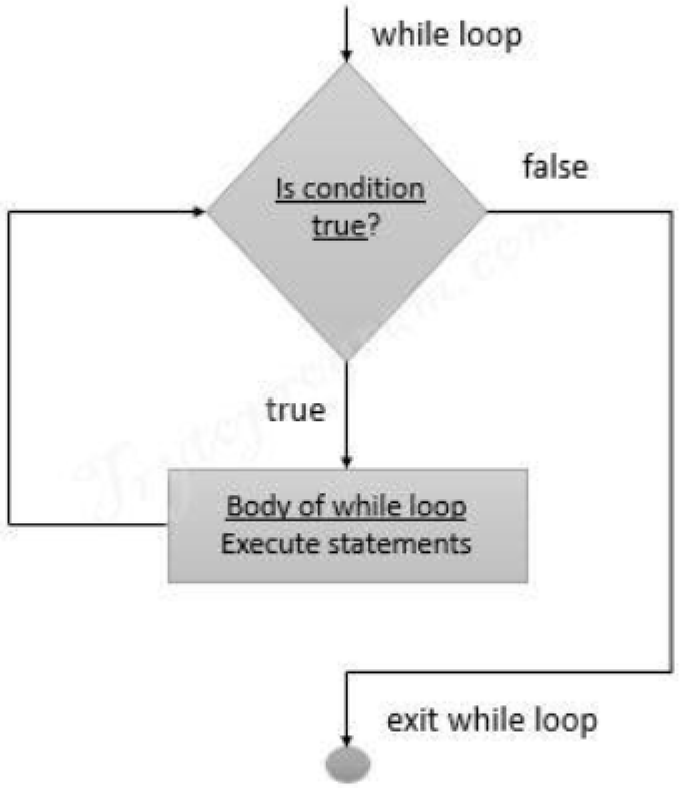


Loops in Python:

Python programming language provides following types of loops to handle looping requirements.

- `while` loops
- `for` loops
- `nested` loops

while loop in python



Example:

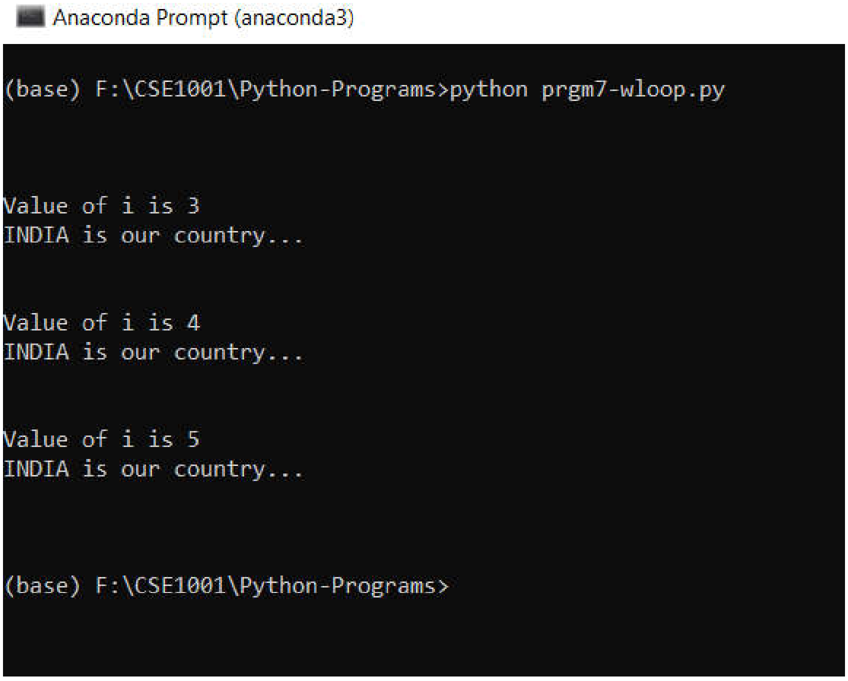
```
i = 1
while i < 6:
  print(i)
  i += 1
```

- Remember to increment i, or else the loop will continue forever.
- The `while` loop requires relevant variables to be ready.
- In this example we need to define an indexing variable, `i`, which we set to 1.

Example:

```
i = 3
print("\n\n")
while i <= 5:
  print("Value of i is", i)
  print("INDIA is our country...")
  print("\n")
  i = i+1
```

Output:

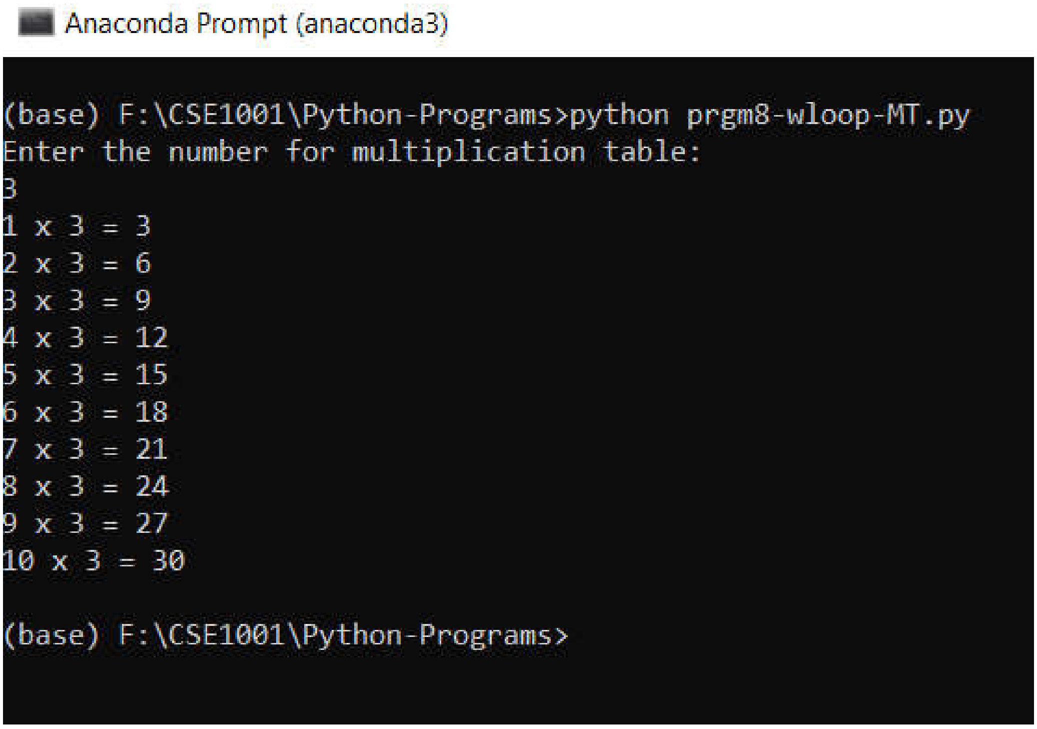


```
(base) F:\CSE1001\Python-Programs>python prgm7-wloop.py



Value of i is 3
INDIA is our country...


Value of i is 4
INDIA is our country...


Value of i is 5
INDIA is our country...



(base) F:\CSE1001\Python-Programs>
```

Example: Printing Multiplication Table

```
#get input from user
num=int(input("Enter the number for multiplication table: \n"))

i=1;
step=10;
while i<=step:
    print(i,'x',num,'=',i*num)
    i+=1
```

Output:



```
(base) F:\CSE1001\Python-Programs>python prgm8-wloop-MT.py
Enter the number for multiplication table:
3
1 x 3 = 3
2 x 3 = 6
3 x 3 = 9
4 x 3 = 12
5 x 3 = 15
6 x 3 = 18
7 x 3 = 21
8 x 3 = 24
9 x 3 = 27
10 x 3 = 30

(base) F:\CSE1001\Python-Programs>
```

The break Statement within while loop

- With the `break` statement we can stop the loop even if the while condition is true:

Example

```
# This program exits from the while loop, when i = 3.

i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

The continue Statement

- With the `continue` statement we can stop the current iteration, and continue with the next:

Example

```
$ Continue to the next iteration, if I = 3.

i = 0
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)

# Note that number 3 is missing in the result
```

Activity:

1) Write a python program using while loop to ad first ‘n’ whole numbers. Read the value for ‘n’ from user.