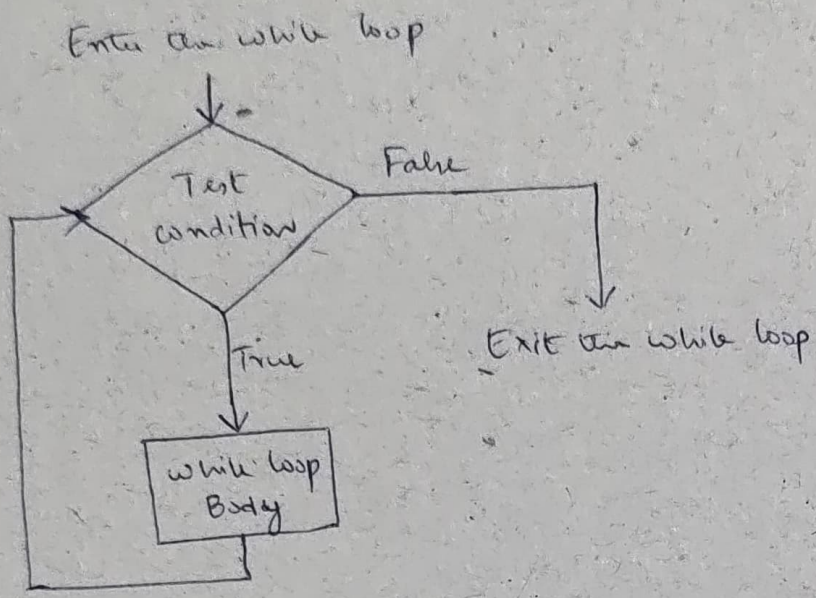
Module 7

Loops in Python

While loop

A while loop is used to execute a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the line immediately after the loop in the program is executed.



- Initializing value
- Loop condition
- Updating value

```
# Initializing value
i = 1

# loop condition
while i <= 6 :
    print ("Welcome")
    #updating value
    i += 1
```

```
Welcome
Welcome
Welcome
Welcome
Welcome
Welcome
```

Print all numbers from 1 to 10 using loop.

```
i = 1
while i <= 10 :
    print (i, end = " ")
    i += 1
```

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

Print all even numbers from 0 to 10

```
i = 0
while i <= 10 :
    if i % 2 == 0 :
        print (i , end = " ")
    i += 1
```

0 2 4 6 8 10

Print sum of all numbers from 1 to 10

```
i = 1
sum = 0
while i <= 10 :
    sum += i
    i += 1
print (sum)
```

55

Range function

- ⊙ range() function allows users to generate series of numbers
- ⊙ It is iterable as well

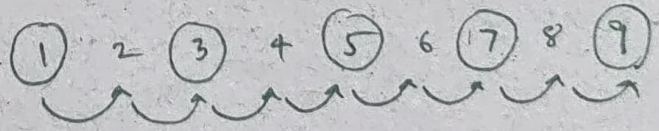
Syntax :

range (start (optional) , stop (required) , step (optional))

Default : 0 — Default : 1

```
list (range (5))
[0, 1, 2, 3, 4]
list (range (1, 5))
[1, 2, 3, 4]
list (range (2, 5))
[2, 3, 4]
list (range (1, 10, 2))
[1, 3, 5, 7, 9]
```

start is included
stop is excluded
included
excluded



```
list (range (10, 1, 1))
[]
list (range (10, 1, -1))
[10, 9, 8, 7, 6, 5, 4, 3, 2]
list (range (-1, 10))
[-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
list (range (10, 0, -2))
[10, 8, 6, 4, 2]
```

Iterator, Iterable and Iteration

Consider the scenario of a Birthday Boy distributing chocolates to all the students in the class. In this scenario, the Birthday boy is the Iterator, All the students in the class room can be thought of Iterable, And the process of Birthday boy distributing chocolates to the students in the class room can be thought of as Iteration.

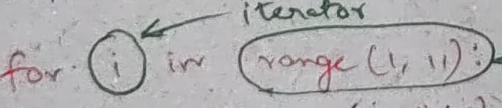
Iterable → An object that can be looped over (class room)

Iterator → An object that represents a stream of data (Birthday boy with chocolates)

Iteration → The process of looping over an iterable object (chocolate distribution)

For loop

A for loop is used for iterating over a sequence (list, tuple, dictionary, set, string). With the for loop, we can execute a set of statements, once for each item in a sequence. The for loop does not require an indexing variable to set beforehand.



1 2 3 4 5 6 7 8 9 10

Print Multiplication Table of 5

```
for i in range(1, 11):
    print(i * 5, end=" ")
```

5 10 15 20 25 30 35 40 45 50

Print Pattern 1

```
# # # #
# # # #
# # # #
# # # #
```

```
for i in range(4):
    print("#", end=" ")
print()
```

```
for i in range(4):
    print("#", end=" ")
print()
```

```
for i in range(4):
    print("#", end=" ")
print()
```

```
for i in range(4):
    print("#", end=" ")
print()
```

⇒

Nested for loop

```
for i in range(4):
    for j in range(4):
        print("#", end=" ")
    print()
```

Print Pattern 2

```
#
# #
# # #
# # # #
```

1 2 3 4

```
for i in range(1, 5):
    for j in range(i):
        print("#", end=" ")
    print()
```

Break, Continue and Pass

These statements are used to alter the flow of a program.

Break : Breaks the flow of program once this condition is hit.

continue: It skips that particular iteration

pass : To avoid Syntax error

Pass

```
for i in range(1, 10):
```

IndentationError : expected an indented block

```
for i in range(1,10):
    pass
```

✓

```
def func():
```

IndentationError : expected an indented block

```
def func():
    pass
```

✓

Continue

```
for i in range(1,10):
    if i==5:
        continue
    print(i, end=" ")
```

1 2 3 4 6 7 8 9

Break

```
for i in range(1,10):
    if i==5:
        break
    print(i, end=" ")
```

1 2 3 4

```
i=1
while i<10:
    if i==5:
        break
    print(i, end=" ")
    i+=1
```

1 2 3 4

Challenges

① Triangle Star - Pattern

Problem description:

Write a function to print the pattern shown in the sample using 'n' given as a parameter, where n is the no. of rows.

Note: There is no any space between consecutive stars

Input format:

The only argument to the function is a number n.

output format:

Triangle Star pattern in string format for each test case

Sample input: 5

Sample output:

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

Sample Explanation:

The first row prints n-1 = 4 spaces and 1 star
The second row prints n-2 = 3 spaces and 3 stars
The third row prints n-3 = 2 spaces and 5 stars
The fourth row prints n-4 = 1 space and 7 stars
The fifth row prints n-5 = 0 space and 9 stars.

```
def star_pattern(n):
    for i in range(1,n+1)
        spaces = ' ' * (n-i)   # n-i spaces
        stars = '*' * (2*i - 1) # 2i-1 stars
        Print (spaces + stars)

star_pattern (5)
```

② Prime finder

Write a function to return True when the given number is a Prime Number, else False.

Input format : An Integer is given

Output format :

True - if Prime
False - if not

Input sample : 5

Output Sample : True

What is a Prime number ?

A Prime number is a whole number greater than 1, with only two factors - 1 and itself. Eg. 2, 3, 5, 7, 11, 13, 17, 19, ...

```
def solve(num)
    if num <=1 :
        return False
    for i in range(2,num):
        if num %i == 0 :
            return False
    return True

solve(5)
True
solve(9)
False
```

③ What is the output of the following Program ?

```
for x in range(10):
    if (x == 6 :
        print(x, end = " ")
print(x)
```

6 9

④ What is the output of the following Program ?

```
for i in range(4):
    if i == 2:
        print(i)
        i = 10
```

2

⑤ What is the output of the following Program ?

```
while i < 20:
    print(i)
    i += 1
```

NameError : name 'i' is not defined.