WHILE LOOP

A while loop in python runs a bunch of code or statements again and again until the given condition is true when the condition becomes false, the loop terminates its repetition.

We have to use the keyword “while”, along with it we have to put a condition in parenthesis and after that, a colon is placed. The condition could be either true or false. Until the condition is true, the loop will keep on executing again and again. If we use a certain sort of condition in our while loop that, it never becomes false then the program will keep on running endlessly, until we stop it by force. So, this kind of mistake in our syntax is known as logical/human error.

To terminate an infinite loop, you can press **Ctrl+C** on your system.

SYNTAX

```
while condition:
        Body of While
```

Examples

```
count = 0
while (count < 9):
    print("The count is : ", count)
    count = count + 1

print("Good bye!")


# -----OUTPUT-----
The count is :  0
The count is :  1
The count is :  2
The count is :  3
The count is :  4
The count is :  5
The count is :  6
The count is :  7
The count is :  8
Good bye!
```

The Infinite Loop

A loop becomes infinite loop if a condition never becomes FALSE. You must use caution when using while loops because of the possibility that this condition never resolves to a FALSE value. This results in a loop that never ends. Such a loop is called an infinite loop.

```
x = 1
while x == 1:  # This shows an infinite loop
    num = input("Enter a number  :")
    print("You entered: ", num)

print("Good bye!")


# -----OUTPUT-----
Enter a number: 3
You entered:  3
Enter a number: 6
You entered:  6
Enter a number: 1
You entered:  1
Enter a number: 9
You entered:  9
Enter a number: 7
You entered:  7
```

Using WHILE Loop with IF and BREAK Statement

Break statement stops the Loop at that Position

```
count = 0
while (count < 9):
    if count == 3:
        break
    print('The count is:', count)
    count = count + 1

print("Good bye!")


# -----OUTPUT-----
The count is: 0
The count is: 1
The count is: 2
Good bye!
```

Using WHILE Loop with IF and CONTINUE Statement

Continue statement skips all the code written below that and goes to next loop.

```
count = 0
while (count < 9):
    count = count + 1
    if count == 4:
        continue
    print('The count is:', count)

print("Good bye!")


# -----OUTPUT-----
The count is: 1
The count is: 2
The count is: 3
The count is: 5
The count is: 6
The count is: 7
The count is: 8
The count is: 9
Good bye!
```

Using WHILE and ELSE

```
i = 1
while i < 4:
    print(i)
    i += 1
else:
    print("i is no longer less than 4")


# -----OUTPUT-----
1
2
3
i is no longer less than 4
```

Nested WHILE Loop

Python programming language allows to use one loop inside another loop which is known as **Nested Loop**.

The syntax for a nested while loop statement in Python programming language is as follows:

```
while expression:
   while expression:
      statement(s)
   statement(s)
```

Example:

```
i = 1
j = 5
while i < 4:
    while j < 8:
        print(i, ",", j)
        j = j + 1
        i = i + 1


# -----OUTPUT-----
1 , 5
2 , 6
3 , 7
```

Questions

1. Print first 10 Natural numbers.
2. Print all the even numbers from 1 to 100.
3. Find sum of N numbers where N is input from User.
4. Check whether number entered by User is Prime or Not.
5. Check whether number entered by User is Armstrong or not.

HW

1. Find Factorial of a number entered by User.
2. Print the first 10 multiples of a number entered by User.
3. Count the number of digits in a number that is entered by User.
4. Write a Program to Reverse a number.