Computational Thinking and Programming - 1

Iterative Statements
(While Loop)

XI

Introduction

Iteration means to do something repeatedly. Iteration is also called loop.

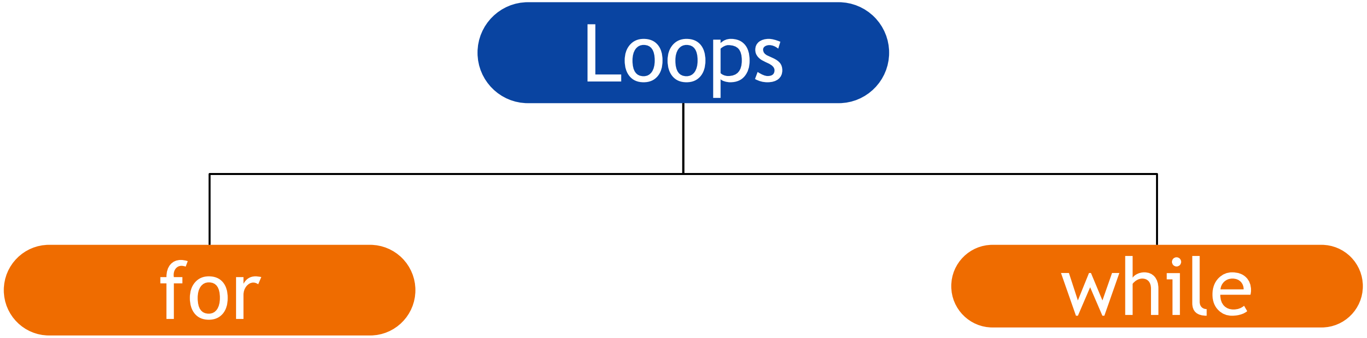
In programming there are many situations where we have to repeat a set of statements.

❖ Suppose we have to input 20 numbers. Should we write input() statement 20 times, or we should have some way so that we write the input() statement once and it is automatically executed 20 times?

This repetition may be for a specified number of times, or as long as a specified condition is True. All programming language provides some iterative or looping constructs which are used to automatically repeat a set of statements.

Types of Loops

There are 2 looping constructs available in Python.



while loop

while is a loop used for repeated execution as long as a specified expression (condition) is true.

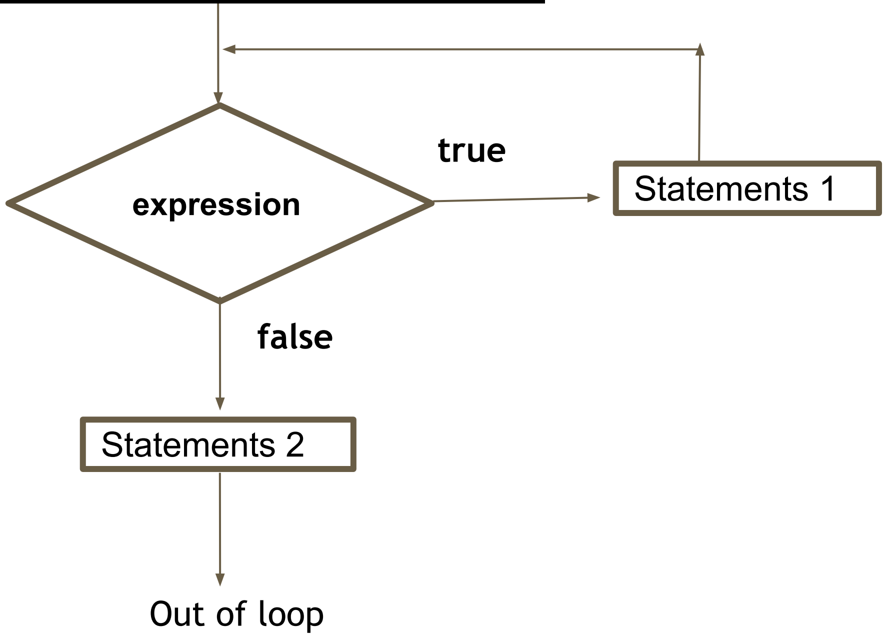
Syntax :

```
while test-expression:
      Statements 1
[else:
      Statements 2]
```

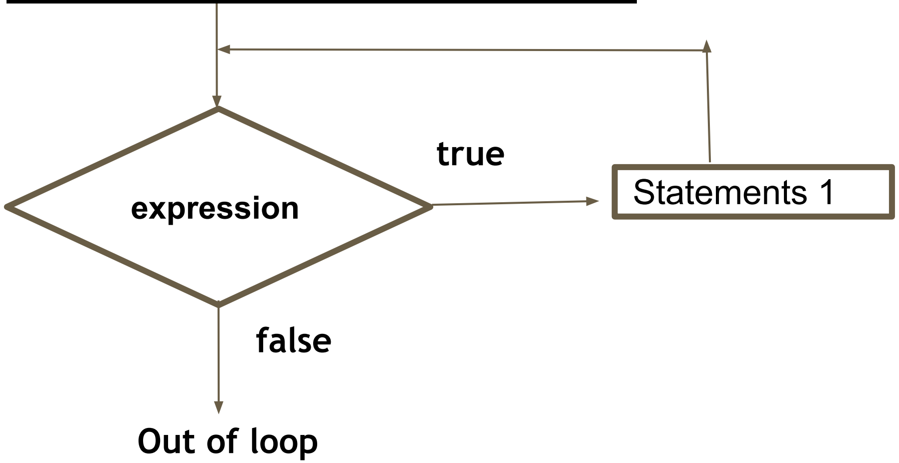
- test expression is a Boolean expression which evaluates to True or False. Such an expression represents a condition.
- else is an optional clause of while loop Execution.

while loop

While loop with else clause



While loop without else clause



Loop Control Elements

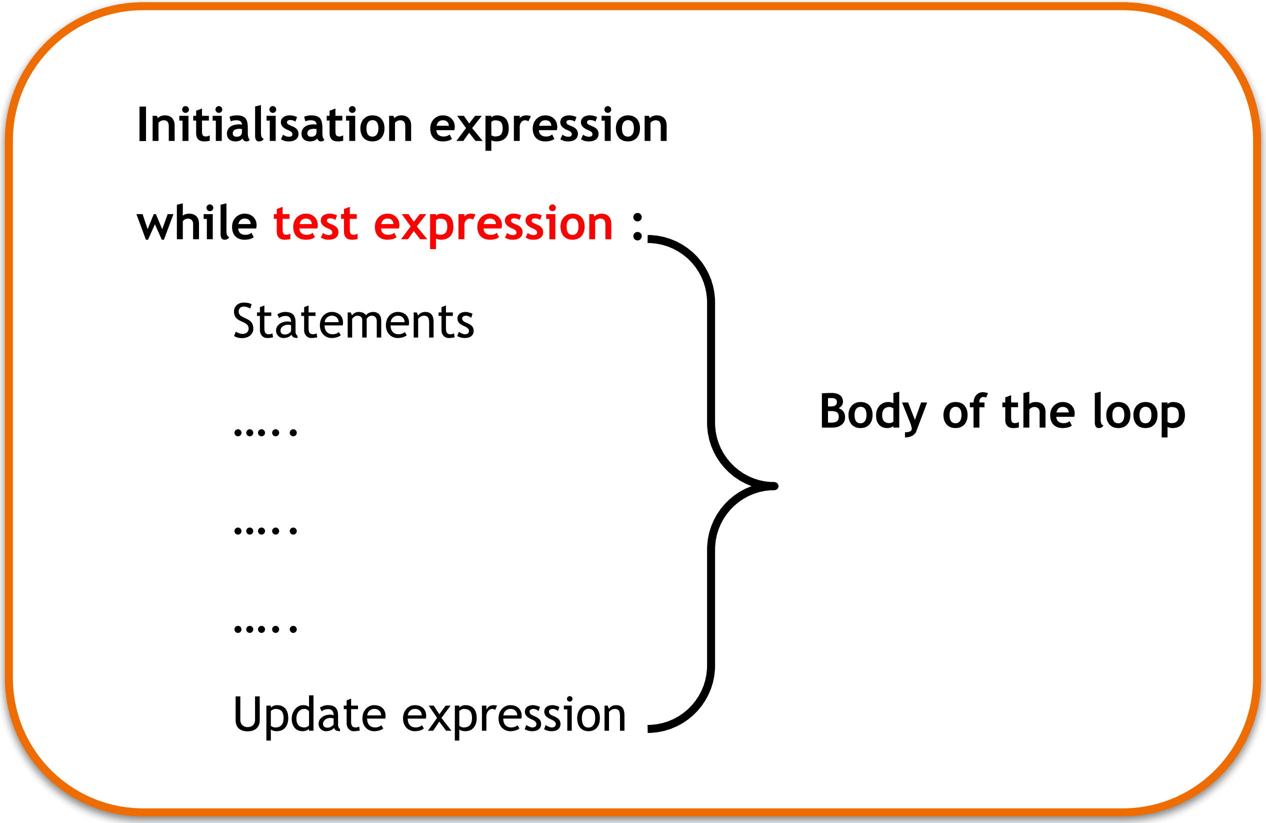
Every while loop has its elements that control and govern its execution. A while loop has four elements that have different purposes:

- **Initialisation expression** : Before entering a loop, the loop variable (also called control variable) must be initialised in an initialisation expression.For a while loop, the initialisation expression is mentioned outside the loop before it starts.
- **Test expression** : It is the condition whose value decides whether the loop body will be executed or not. If the test expression evaluates to True, the loop body gets executed, otherwise the loop is terminated.

Loop Control Elements

- **Body of the loop** :The statements that have to be executed repeatedly form the body of the loop. The body of the loop may be a simple statement or a compound statement.
- **Update expression** : The statement that changes the value of the loop variable is called an update expression. This expression is mentioned inside the body of the loop.

while loop



```
Initialisation expression

while test expression :

        Statements

        …..

        …..

        …..

        Update expression
```

Body of the loop

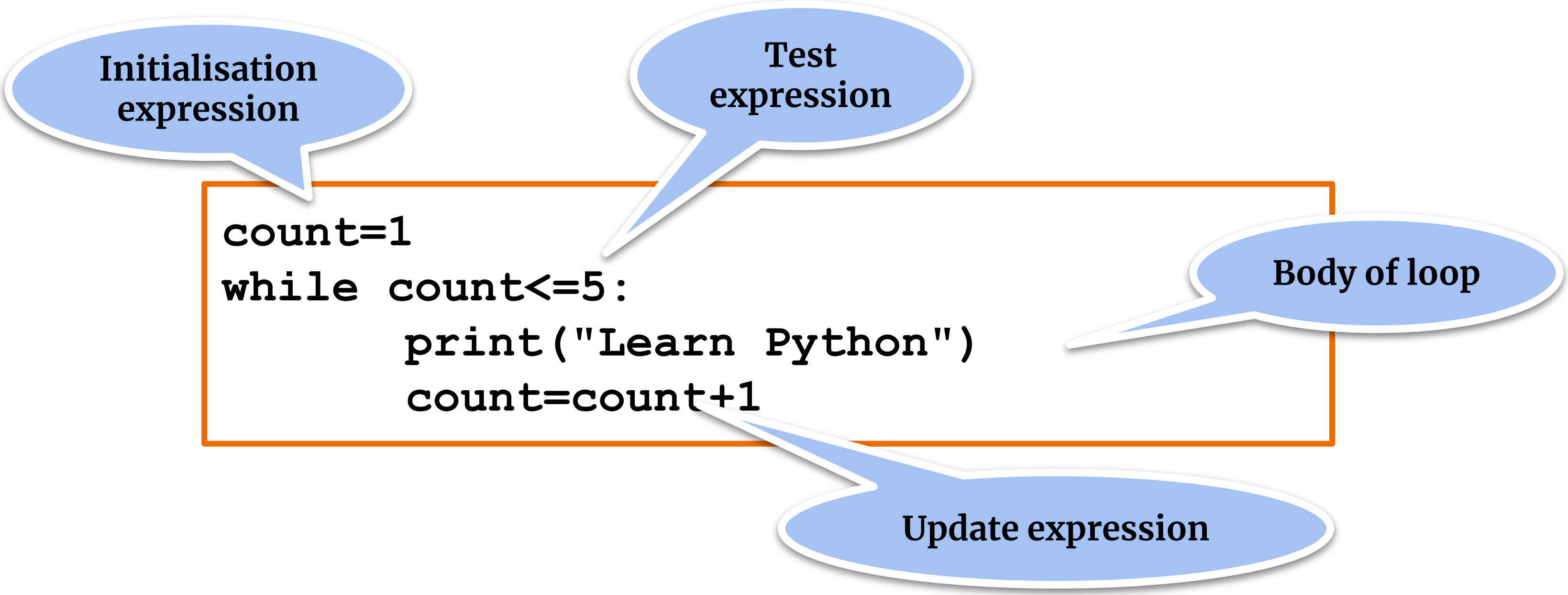
Some important points:

- In a while loop, the loop control variable should be initialized before the loop begins as an uninitialised variable cannot be used in the test condition (expression)

- The loop variable must be updated inside the body of the loop in a way that after sometime the test condition becomes false otherwise the loop will become an endless loop or infinite loop.

- While loop is also called **entry-controlled loop** as the entry into the loop is controlled by testing condition.

Example: Code to print a message 5 times on



```
count=1
while count<=5:
        print("Learn Python")
        count=count+1
```

Example : Code to print first 10 natural

```
num=1
print("First 10 Natural Numbers")
while num<=10:
    print(num)
    num=num+1
```

```
First 10 Natural Numbers
1
2
3
.
.
.
10
```

Question 1

Write a program to print all even numbers from 1 to 10 and find their sum too.

```
num=2
s=0
print("Even Numbers are :")
while num<=10:
    print(num)
    s=s+num
    num=num+2
print("Sum of even numbers :",s)
```

```
Even numbers are:
2
4
6
8
10
Sum of even numbers
:40
```

Question 2

Write a program to print table of number n.

```
num=int(input("Enter the number:"))
print("Table of number :")
i=1
while i<=10:
    print(num,"x",i,"=",num*i)
    i=i+1
```

```
Enter the number: 6
Table of number :
6x1=6
6x2=12
6x3=18
.
.
.
6x10=60
```

Question 3

Write a program input a number and then find the factorial of a number.

Factorial of number n is calculated as : $n!=n\times(n-1)\times(n-2)\ldots\times2\times1$

```
num=int(input("Enter the number:"))
i=1
f=1
while i<=num:
    f=f*i
    i=i+1
print("Factorial of number :",f)
```

```
Enter the number:5
Factorial of number :
120
```

Programs

1. Print all the natural numbers from m to n.
2. Input a number n and then print all natural numbers from n to 1.(in reverse order)
3. Input a number n and then print all the odd numbers from 1 to n along with their sum.
4. Input a number n and then print all its factors.
5. Input a number and then print the reverse of the number.
6. Input a number and check if it a palindrome number or not
7. Input a number and then print if the number is an armstrong number or not .
   Example : (153 = 13+53+33)
8. Input a number and then print if the number is a perfect number or not.
   Example : (6 =1+2+3)
9. Input the number and then print the Fibonacci series till n terms
10. Input a number and then check if it is a prime number or not.

Question 1

Write a program to print all the natural numbers from m to n.

```
m=int(input("Enter the number m:"))
n=int(input("Enter the number n:"))
print("Natural numbers are:")
while m<=n:
    print(m)
    m=m+1
```

```
Enter the number m: 10
Enter the number n: 16
Natural numbers are:
10
11
12
13
14
15
16
```

Question 2

Write a program to print all the natural numbers from m to 1.(Reverse Order)

```
m=int(input("Enter the number m:"))
print("Natural numbers are:")
while m>=1:
    print(m)
    m=m-1
```

```
Enter the number m: 7
Natural numbers are:
7
6
5
4
3
2
1
```

Question 3

Write a program to input a number n and then print all the odd numbers from 1 to m along with their sum.

```
m=int(input("Enter the number m:"))
print("Natural numbers are:")
sum=0
i=1
while i<=m:
    print(m)
    sum=sum+m
    m=m+2
print("Sum :",sum)
```

```
Enter the number m: 5
Natural numbers are:
1
3
5
Sum : 9
```

Question 4

Write a program to input a number n and then print all its factors.

```
n=int(input("Enter the number n:"))
i=1
print("Factors are:")
while i<=n:
    if n%i==0:
        print(i)
    i=i+1
```

```
Enter the number m: 15
Factors are:
1
3
5
15
```

Question 5

Write a program to input a number n and then print the sum of all its factors.

```python
m=int(input("Enter the number m:"))
i=1
sum=0
print("Factors are:")
while i<=m:
    if m%i==0:
        print(i)
        sum+=i
    i=i+1
print("The sum of the factors is :", sum)
```

```
Enter the number m: 15
Factors are:
1
3
5
15
The sum of the factors is
28
```

Question 5

Write a program to input a number and find the reverse of the number.

```
m=int(input("Enter the number:"))
rev=0
while m>0:
    dig=m%10
    rev=rev*10+dig
    m=m//10
print("Reverse of number :",rev)
```

```
Enter the number: 2309
Reverse Number : 9032
```

Explanation :

Number n : 2309, rev=0			
Iteration	dig=n%10	rev=rev*10+dig	n
I	9	9	230
II	0	90	23
III	3	903	2
IV	2	9032	0

Reverse Number

Test condition: while n > 0:

Question 5

Write a program to input a number and check if it a palindrome number or not.

A palindrome is a word or number which reads the same backward as forward.
For example: 919 is a palindrome

```
num = int(input("Enter a number: "))
rev = 0
t = num
while t > 0:
   digit = t % 10
   rev=rev*10+digit
   t //= 10
if num == rev:
   print(num,"is a Palindrome")
else:
   print(num,"is not a palindrome")
```

```
Enter a number:
4224
4224 is a palindrome
```

Question 6

Write a program to input a number and then print if the number is an armstrong

Armstrong number is a number that is equal to the sum of cubes of its digits. Eg:(153 = $1^3+5^3+3^3$)

```
num = int(input("Enter a number: "))
sum = 0
t = num
while t > 0:
   digit = t % 10
   sum += digit ** 3
   t //= 10
if num == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")
```

```
Enter a number: 153
153 is an Amrstrong
number
```

Question 7

Write a program to print fibonacci series till value n

The Fibonacci Sequence is the series of numbers:
0, 1, 1, 2, 3, 5, 8, 13

```
num = int(input("Enter a number: "))
a=0
b=1
print(a,end=',')
print(b,end=’,’)
c=a+b
while c <=num:
    print(c,end=',')
    a=b
    b=c
    c=a+b
```

```
Enter a number: 10
0,1,1,2,3,5,8
```

Question 7

Write a program to print fibonacci series for n number of terms

```
nterms = int(input("Enter the number of terms : "))
n1, n2 = 0, 1
count = 0
if nterms <= 0:
   print("Please enter a positive integer")
elif nterms == 1:
   print("Fibonacci series upto",nterms," :")
   print(n1)
else:
   print("Fibonacci series :")
   print(n1)
   print(n2)
   while count < nterms-2:
       n3 = n1 + n2
       print(n3)
       n1 = n2
       n2 = n3
       count =count+1
```

```
Enter a number:  6
0
1
1
2
3
5
```

Question 8

Write a program to check whether number is prime or not.

Prime number is one which is only divisible by 1 or itself

```
num = int(input("Enter the number  :"))
flag=0
i=2
while i<num:
    if num % i==0:
        flag=1
        break
    i=i+1
if flag==0:
    print("Prime Number")
else:
    print("Not a Prime Number")
```

```
Enter a number: 11
Prime Number
```

while loop with else clause

The else part is executed if the condition in the while loop evaluates to False.

Example :

```
counter=0
while counter<3:
    print(“Inside loop”)
    counter=counter+1
else:
    print(“Inside else”)
```

Output :

```
Inside loop
Inside loop
Inside loop
Inside else
```

Nested while loop

Nested loop is a loop inside another loop.

Example :

```
n=int(input('Enter the no of lines:'))
i=1
while i<=n:
    j=1
    while j<=i:
        print(j,end='')
        j=j+1
    print( )
    i=i+1
```

Output :

```
Enter the no of lines:5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Nested while loop

```
n=int(input('Enter the no of lines:'))
i=1
while i<=n:
    j=1
    while j<=i:
        print(j,end='')
        j=j+1
    print( )
    i=i+1
```

Value of i	Value of j
1	1
2	1 2
3	1 2 3
4	1 2 3 4
5	1 2 3 4 5

Programs

1. Write a program to print the following patterns :

a)

```
1 2 3 4 5

1 2 3 4

1 2 3

1 2

1
```

b)

```
1

2  2

3   3   3

 4    4   4   4

  5    5    5   5   5
```

c)

```
*

*  *

*    *    *

*    *    *    *

*    *    *     *     *
```

Program to generate factorial of all natural numbers till n.

```
n=int(input("Enter the value of n :"))
i=1
while i<=n:
    j=i
    fact=1
    while j>=1:
        fact=fact*j
        j-=1
    print("Factorial of number ",i," is ",fact)
    i=i+1
```

```
Enter the value of n :5
Factorial of number 1 is 1
Factorial of number 2 is 2
Factorial of number 3 is 6
Factorial of number 4 is 24
Factorial of number 5 is 120
```

Programs

1. Program to print all the prime numbers till n.
2. Program to print all armstrong numbers which lie between m and n.
3. Program to print all palindrome numbers till n.
4. Program to print the factorial of all numbers from 1 till n where n is input by the user

Program to print all the prime numbers till n.

```python
n=int(input("Enter the value of n :"))
print("Prime Numbers are :")
i=1
while i<=n:
    flag=0
    j=2
    while j<i:
        if i % j==0:
            flag=1
            break
        j=j+1
    if flag==0:
        print(i)
    i=i+1
```

```
Enter the value of n :5
Prime numbers are:
2
3
5
```

Print Armstrong numbers between m and n.

```
m=int(input("Enter the value of m :"))
n=int(input("Enter the value of n :"))
print("Armstrong Numbers are :")
i=m
while i<=n:
    sum = 0
    t = i
    while t > 0:
        digit = t % 10
        sum += digit ** 3
        t //= 10
    if i == sum:
        print(i)
    i=i+1
```

```
Enter the value of m :5
Enter value of n: 200
Armstrong numbers are:
153
```

Print all palindrome numbers till n.

```
m=int(input("Enter the value of m :"))
print("Palindrome Numbers are :")
i=1
while i<=m:
    rev = 0
    t = i
    while t > 0:
        digit = t % 10
        rev=rev*10+digit
        t //= 10
    if i == rev:
        print(i)
    i=i+1
```

```
Enter the value of m :20
Palindrome Numbers are:
1
2
3
4
5
6
7
8
9
11
```

Find the errors in the following code :

Code 1:

```
a=int(input("Enter a
number: ")
while a%5!=0
    if (a%2>0):
        print(a,sep="--)
    a+=1
else: print(a)
```

Code 2:

```
x=123045
while x%10:
    x=//10
    print(x,sep="--")
    else print(a)
```

Solution:

```
a=int(input("Enter a number: ")
while a%5!=0
    if (a%2>0):
        print(a,sep="--)
    a+=1
else: print(a)
```

Missing closing bracket

Missing colon

Missing double quotes

```
x=123045
while x%10:
    x=//10
    print(x,sep="--")
    else print(a)
```

Wrong assignment operator

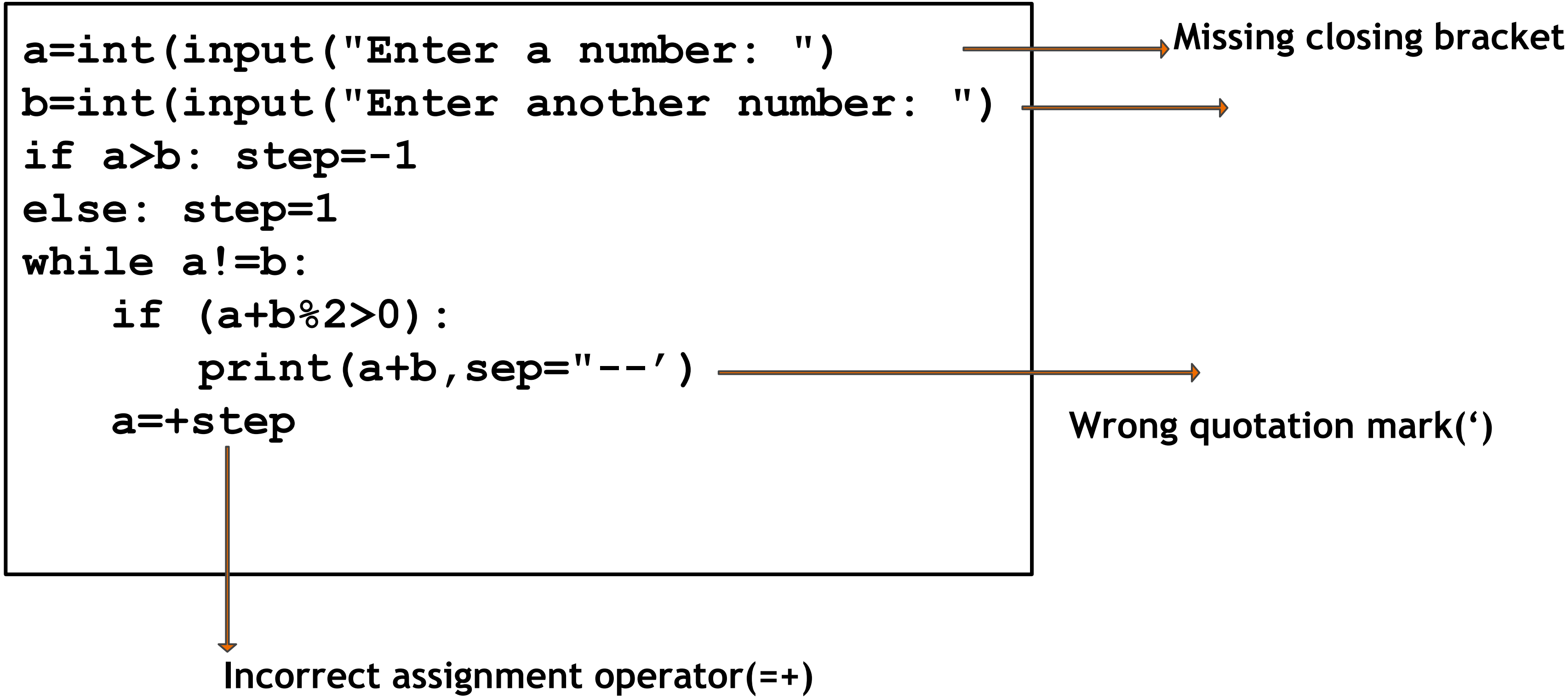
a is not defined

Wrong indent and Missing colon

Find the errors in the following code :

```
a=int(input("Enter a number: ")
b=int(input("Enter another number: ")
if a>b: step=-1
else: step=1
while a!=b:
    if (a+b%2>0):
        print(a+b,sep="--’)
    a=+step
```

Solution



```
a=int(input("Enter a number: ")
b=int(input("Enter another number: ")
if a>b: step=-1
else: step=1
while a!=b:
    if (a+b%2>0):
        print(a+b,sep="--’)
    a=+step
```

Missing closing bracket

Wrong quotation mark(‘)

Incorrect assignment operator(=+)

Find the output of the following code:

```
a=int(input('Enter a number:'))
b=int(input('Enter a number:'))
start,end=a,b
if a>b:
    start,end=b,a
i=start
while i<=end:
    if (i%start==0):
        print(i,end='-')
    i+=1
print(i)
```

Find the output for :

1. a,b=4,10
2. a,b=2,-4

Solution:

```
a=int(input('Enter a number:'))
b=int(input('Enter a number:'))
start,end=a,b
if a>b:
    start,end=b,a
i=start
while i<=end:
    if (i%start==0):
        print(i,end='-')
    i+=1
print(i)
```

Find the output for :

1. a,b=4,10
2. a,b=2,-4

```
Enter a number : 4
Enter a number : 10
4-8-11
```

```
Enter a number : 2
Enter a number : -4
-4-0-3
```

Give output of the following code:

```
a=10
b=a-6
if a>b:
    step=-1
else: step=1
while a!=b:
    if (a+b%2>0):
        print(a+b,sep='--')
    a+=step
```

Solution:

```
a=10
b=a-6
if a>b:
    step=-1
else: step=1
while a!=b:
    if (a+b%2>0):
        print(a+b,sep='--')
    a+=step
```

```
14
13
12
11
10
9
```

Give output :

```
a=int(input("Enter the value of n : "))
while a%5!=0:
    if a%2>0:
        print(a,sep="-")
    a+=1
else:print(a)
```

```
Find the output for :

10
-6
```

Solution:

```
a=int(input("Enter the value of n : "))
while a%5!=0:
    if a%2>0:
        print(a,sep="-")
    a+=1
else:print(a)
```

Find the output for :
10
-6

```
Enter value of n : 10
10
```

```
Enter value of n : -6
-5
```

Give output :

```
x=123045
while x%10:
    x//=10
    print(x,sep="--")
else:print(x)
```

Solution

```
x=123045
while x%10:
    x//=10
    print(x,sep="--")
else:print(x)
```

```
12304
1230
1230
```

Give output of the following code:

```
choice="y"
while choice=="y":
    A = int(input("A : "))
    B = int(input("B: "))
    if A >= B:
        print("Stop!")
        break
    C = A
    while C <= B:
        print(C,end = "*")
        C+=1
    print()
   choice=input("Add more")
```

INPUT
A = 10
B = 20

OUTPUT:

```
10*11*12*13*14*15*16*17*18*19*20*
```

Rewrite the code using for loop:

```
num=10

sum=0

count=0

while num > 0 :

    count += 1

    sum += num

    num -= 2

    if count == 10 :

        print (sum/float(count))

        break
```

Rewrite the code using for loop:

```
num=10

sum=0

count=0

while num > 0 :

    count += 1

    sum += num

    num -= 2

    if count == 10 :

        print (sum/float(count))

        break
```

```
num=10

sum=0

count=0

for i in range(num,0,-2):

    count+=1

    sum+=i

    if count==10:

        print(sum/float(count))

        break
```

Rewrite the code using for loop:

```
i = 100
while (i > 0)  :
    print (i)
    i -= 3
```

Rewrite the code using for loop:

```
i = 100

while (i > 0)  :

     print (i)

     i -= 3
```

```
for i in range(100, 0, -3) :

   print (i)
```

Rewrite the code using while loop:

```
for i in range(4) :
    for j in range(5):
        if i + 1 == j or j + 1 == 4 :
           print ("+", end = ' ')
    else :
        print ("o", end = ' ')
print()
```

Rewrite the code using while loop:

```
for i in range(4)  :
    for j in range(5):
        if i + 1 == j or j + 1 == 4  :
            print ("+", end = '  ')
        else  :
            print ("o", end = '  ')
print()
```

```
i = 0
while i < 4:
    j = 0
    while j < 5:
        if i + 1 == j or j + 1 == 4 :
            print ("+", end = ' ')
        else :
            print ("o", end = ' ')
        j += 1
    i += 1
print()
```

Rewrite the code using while loop:

```
num=5
min = 0
max = num
if num < 0 :
    min = num
    max = 0
    for i in range(min, max + 1):
        sum += i
```

Rewrite the code using while loop:

```
num=5
min = 0
max = num
if num < 0 :
    min = num
    max = 0
    for i in range(min, max + 1):
        sum += i
```

```
num=5
min = 0
max = num
if num < 0 :
    min = num
    max = 0
    i = min
    while i <= max:
        sum += i
        i += 1
```

Happy Learning

Thank you!!!