

## Python while loop

the while loop in python is used to iterate over a block of the code as long as the test expression(condition)is true.

### Syntax:

```
while text_expression:
```

```
    Body of while
```

the body of the loop is entered only if the test\_expression evaluates to true.

After one iteration, the test expression is checked again.

This process continues until the test\_expression to false.

```
# Find product of all numbers present in list.
lst = [10, 20, 30, 40]
product = 1
index = 0
print(len(lst))
while index < len(lst): #
    product *= lst[index] # product = product * lst[index] product =
200 * 30 = 6000
    index = index + 1

print("product is {}".format(product))
#print(" product is",product)

4
product is 240000
```

## While loop with else

same as that of for loop, we can have an optional else block with while loop as well.

The else part execute if the condition in the while loop evaluates to false. The while loop can terminated with a break statment.

In such case, the else part is igonred. Hence, a while loop's else part runs if no break occurs and the condition is false.

### Syntax:

```
while text expression:
    body of while
else:
    body of else statement
```

```
numbers = [1, 2, 3, 4, 5]
```

```
# iteration over the list
```

```
index = 0
```

```
while index < len(numbers): # 5 < 5
```

```
    print(numbers[index])
```

```
    index +=1
```

```
else:
```

```
    print("No item left in list")
```

```
1
2
3
4
5
No item left in list
```

### Python program to check give number is prime number or not

```
num = int(input("Enter the number: "))
```

```
a= 0
```

```
i = 2
```

```
while i < num: #2<144
```

```
# num =
```

```
6 i = 2,3,4,5,---,143
```

```
    if num % i == 0: # 6 % 2 == 0
```

```
        a = 1
```

```
        #print(" {} is divisible by {}".format(num,i))
```

```
        break
```

```
    i +=1 # i = 5
```

```
if a==0:
```

```
    print("{} is a prime number".format(num))
```

```
else:
```

```
    print("{} is not a prime number".format(num))
```

```
Enter the number: 144
```

```
144 is divisible by 2
```

```
144 is not a prime number
```

## Python for loop

The for loop in python used to iterate over a sequence(list, tuple, string) or other objects. iteration over a sequence is called traversal.

### Syntax:

```
for element in sequence:
```

```
    Body of for
```

Here, element is the variable that takes the value of the item inside the sequence on each iteration.

loop continues until we reach the last item in the sequence.

```
# find product of all numbers present in a list
```

```
lst = [10, 20, 30, 40]
product = 1
```

```
#iteration over list
```

```
for ali in lst: # ele = null in lst
    product *= ali # product = product * harsh = 200 * 30 = 6000
```

```
print(" product is: {}".format(product))
print(id(lst))
```

```
product is: 240000
2284117387976
```

## range() fuction

we can generate a sequence of number using range() function. range(10) will generate numbers form 0 to 9 (10 numbers)

we can also define that start, stop and step size as range(start, stop, step size). step size defaults to 1 if not provided.

This function does not store all the values in memory, it would be inefficient. So it remembers the start, stop, step size and generates nextnumber on the go.

```
#print range of 10
```

```
for a in range(25): #[0,1,2,3,4,5...,24] a =1 in range
    print(a)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

```
#print range of numbers from 1 to 15 with step size of 2
for i in range(0,1,): # step mean 2 .... every second element
    [1,2,3,4,5...,24] step -1
    print(i)
```

```
File "<ipython-input-25-389c7796c4bf>", line 2
    for i in range(,1,): # step mean 2 .... every second element
    [1,2,3,4,5...,24] step -1
    ^
```

SyntaxError: invalid syntax

```
lst = ["harsh", "sumit", "manav", "raj", "rv"]
for i in range(len(lst)): # range [0,1,2,3,4,5] index 1
    print(lst[i])
```

```
print(lst[-1:-len(lst):-1])
```

```
harsh
sumit
manav
raj
rv
['rv', 'raj', 'manav', 'sumit']
```

```
lst = ["harsh", "sumit", "manav", "raj", "rv"]
for ele in lst:
    if ele == "himanshu":
        print(ele)
```

### for loop with else

```
number = [1, 2, 3, 4, 5, 6, 7]
for item in number:
    print(item)
else:
    print("no item left in the list")
```

```
1
2
3
4
5
6
7
no item left in the list
```

### Python program to display all prime numbers within an interval

```
in1 = 500
in2 = 1000
print(" prime number between {0} and {1} are :".format(in1,in2))
```

```
for num in range (in1,in2+1): #default step size is 1
    range[10,11,12,13,...,1001] num = 10
    if num > 1:                # 11>1
        a = False             # a = False
        for index in range (2, num): # index in
            range[2,3,4,5,6,7,8,9,10]
                if num % index == 0:      # 11 % 2
                    a = True
        if not a:      # not a = not False = True
            print(num)
```

```
prime number between 500 and 1000 are :
503
509
521
523
541
547
557
563
569
571
```

577  
587  
593  
599  
601  
607  
613  
617  
619  
631  
641  
643  
647  
653  
659  
661  
673  
677  
683  
691  
701  
709  
719  
727  
733  
739  
743  
751  
757  
761  
769  
773  
787  
797  
809  
811  
821  
823  
827  
829  
839  
853  
857  
859  
863  
877  
881  
883  
887  
907

911  
919  
929  
937  
941  
947  
953  
967  
971  
977  
983  
991  
997

```
LOC(j)
a = [1][2][3]----[j]----[n]
LOC(j)= baseaddress + (j - 0)*type of array
```

```
l=0 , k= 1 , j= 0;
m = l++ && k++ && j++;
```

```
num = int(input("Enter an integer"))
n = 50
i = 1
while(i<=n):
    print("{} * {} = {}".format(num,i,num*i))
    i+=1
```

```
t = (1,2,3,[4,5,6])
print(t,type(t))
t[3][1]= 7
print(t)
```

```
(1, 2, 3, [4, 5, 6]) <class 'tuple'>
(1, 2, 3, [4, 7, 6])
```