**Loops**

- Repeating the same task
- For example we want to do tax calculation
- Tax calcilation has 4 lines of code
- I want to apply the tax calculations to my 100 employees
- How many lines required : 400
- But if I keep the 4 lines in form of function
- Then only one line can use for 100 members
- The making one line : Functions
- we are applying that one line for all 100 members : Loop

- Generally we have two types of loops
  - For loop
  - while loop
- Any loop we required three things
  - Intial point : start point to enter into the loop
  - Increment or decrement : Number of times iterate the loop
  - Condition to stop : we want to stop the loop based on specific condition
- In for loop all these 3 are in a single line
- In while loop all these 3 are 3 different lines

**For loop**

$pattern - 1$

**range(stop)**

- for loop use range data type
- inside range we can keep some values
- suppose if you keep only one value
- start= 0 by default

- increment by 1
- stop value= the given value -1
- range(3)
    - start=0
    - increment=1
    - stop= last-1= 3-1=2
    - 0 1 2

**python index start with zero**

In [1]:

```
# syntax

for i in range(3):
    print(i)
```

```
0
1
2
```

In [2]:

```
print(0)
print(1)
print(2)
print(4)

print(i)
```

```
0
1
2
4
```

In [ ]:

```
for i in range(4):
    print(i)
# step-1  i=0  ===== print(0)
# step-2  i=1  ===== print(1)
# step-3  i=2  ===== print(2)
# step-4  i=3  ===== print(3)
```

In [10]:

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

```
0 1 2 3
```

In [9]:

```
print(0,end=' ')
print(1,end=' ')
print(2,end=' ')
print(4)

print(i,end=' ')
```

```
0 1 2 4
```

In [11]:

```
# print hello 5 times
print('hello')
print('hello')
print('hello')
print('hello')
print('hello')
```

```
hello
hello
hello
hello
hello
```

In [12]:

```
for i in range(5):
    print('hello')
```

```
hello
hello
hello
hello
hello
```

In [ ]:

```
# wap ask the use get a number ranndomly between 1 to 100
#       print the square of the number
#   ask the user repeat this task for three times
```

In [14]:

```
import random
num=random.randint(1,100)
print(f"the square of {num} is {num*num}")

import random
num=random.randint(1,100)
print(f"the square of {num} is {num*num}")

import random
num=random.randint(1,100)
print(f"the square of {num} is {num*num}")
```

```
the square of 2 is 4
the square of 32 is 1024
the square of 82 is 6724
```

In [15]:

```
import random
for i in range(3):
    num=random.randint(1,100)
    print(f"the square of {num} is {num*num}")
```

```
the square of 8 is 64
the square of 89 is 7921
the square of 96 is 9216
```

$$pattern-2$$

**range(start,stop)**

- Whenever we see the above pattern
- start= start
- increment by 1
- last=stop-1

- range(5,10)
    - start=5
    - increment=1
    - last= 10-1=9
    - 5,6,7,8,9

In [16]:
```
for i in range(5,10):
    print(i)
```

```
5
6
7
8
9
```

In [17]:
```
# wap ask the user print the square of the numbers
# from 1 to 5
for i in range(1,6):
     print(f"the square of {i} is {i*i}")
```

```
the square of 1 is 1
the square of 2 is 4
the square of 3 is 9
the square of 4 is 16
the square of 5 is 25
```

In [20]:
```
import random
for i in range(1333,1336):
    num=random.randint(1,100)
    print(f"the square of {num} is {num*num}")
```

```
the square of 99 is 9801
the square of 6 is 36
the square of 7 is 49
```

In [22]:
```
# wap ask the use get a number from keyboard
#   find that number even or odd
# repeat this 5 times

for i in range(5):
    num=eval(input('enter a number:'))
    if num%2==0:
        print(f"the {num} is an even")
    else:
        print(f"the {num} is an odd")
```

```
the 1 is an odd
the 2 is an even
the 3 is an odd
the 4 is an even
the 5 is an odd
```

In [23]:
```
num= eval(input("enter the number:"))
if num%2==0:
    for i in range(5):
         print(f"the number is even {num}")
```

```
else:
    print("number is odd")
```

```
the number is even 20
the number is even 20
the number is even 20
the number is even 20
the number is even 20
```

$pattern - 3$

**range(start,stop,step)**

- start= start only
- step value
    - if step value is postive
        - postive step means postive direction
        - increment= step
        - last= stop-1
    - if step value is negative
        - negative step means reverse direction
        - decrement = step
        - last= stop+1

In [25]:

```
for i in range(1,11,2):
    print(i,end=' ')

# start=1
# direction= step =+2
# increment =2
# last= stop-1=11-1=10

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

```
1 3 5 7 9
```

In [26]:

```
for i in range(1,11,-2):
    print(i,end=' ')

# start=1
# negative = step = -2
# last = stop+1= 11+1=12
```

In [27]:

```
for i in range(11,1,-2):
    print(i,end=' ')

# start=11
# dire = negative = -2
# last=stop+1= 1+1=2
# 11  10  9  8  7  6  5  4  3  2
# 11      9     7     5     3
```

```
11 9 7 5 3
```

In [28]:
```
for i in range(-11,1,-2):
    print(i,end=' ')

# start=-11
# direction= negative step=-2  decrement
# last= stop+1=1+1=2
```

In [29]:
```
for i in range(-11,-1,2):
    print(i,end=' ')
```

```
-11 -9 -7 -5 -3
```

In [ ]:
```
range(3,15,3) # P
range(3,15,-3) # Np
range(3,-15,3) # NP
range(-3,15,3) # P
range(3,-15,-3) # P
range(-3,15,-3) # NP
range(-3,-15,-3) # P
range(15,3,3) # NP
range(15,3,-3) # P
range(15,-3,3) # NP
range(-15,-3,-3) # NP
```

In [1]:
```
range(-22,22,-2)
```

Out[1]:
```
range(-22, 22, -2)
```

In [ ]:
```
#1) wap ask the use get a number ranndomly between 1 to 100
#       print the square of the number
#   ask the user repeat this task for three times
import random
for i in range(3):
    num=random.randint(1,100)
    print(f"the square of {num} is {num*num}")
```

In [ ]:
```
#2) wap ask the user print the square of the numbers
# from 1 to 5
for i in range(1,6):
     print(f"the square of {i} is {i*i}")
```

In [ ]:
```
#3) wap ask the use get a number from keyboard
#   find that number even or odd
# repeat this 5 times

for i in range(5):
    num=eval(input('enter a number:'))
    if num%2==0:
        print(f"the {num} is an even")
    else:
        print(f"the {num} is an odd")
```

In [6]:

```
#4) Wap: Print the 7the table
# 7 x 1 = 7
# 7 x 2 = 14
# 7 x 10 = 70

# 7 x i = ans
for i in range(1,11):
    #print(7*i)
    print(f"7x{i}={7*i}")
```

```
7x1=7
7x2=14
7x3=21
7x4=28
7x5=35
7x6=42
7x7=49
7x8=56
7x9=63
7x10=70
```

In [10]:

```
def table():
    num=eval(input('Enter the number to display the table:'))
    for i in range(1,11):
        print(f"{num}x{i}={num*i}")

table()
```

```
19x1=19
19x2=38
19x3=57
19x4=76
19x5=95
19x6=114
19x7=133
19x8=152
19x9=171
19x10=190
```

In [ ]:

```
# Code
# Type the code
# 5 qns
# till if-else
# online offline
# write the exam
```

In [12]:

```
# 5) WAP to find the divisiors of a given number : 75
# What is mean by divisors
# If you dived a number with another number the remineder =0
# Idea
# Itearte the loop with given number
# For example if you want divisors of number =10
# start=1   end with 10
# So there are 1 to 10 numbers
# every number divide with the itartor number
# Find the reminder
# if reminder =0 then print that number

# step-1: num= eval()
# step-2: for i in range(1,num+1):
```

```
# step-3:     if <reminder>:
# step-4:         print(i)

num=eval(input("enter the number:"))
for i in range(1,num+1):
    if num%i==0:
        print(i)
```

```
1
3
5
15
25
75
```

In [ ]:

```
# 6) wap ask the to get sum of first 10 natural numbers
# 1+2+3+4+5+6+7+8+9+10= 55

0+1=1
1+2=3
3+3=6
6+4=10
10+5=15
15+6=21
21+7=28
28+8=36
36+9=45
45+10=55

{sum}+i={sum}
```

**Summation wrapper**

- whenever we do summation using loop
- always keep summ=0 at starting of the code
- second line iterate the for loop
- Third line inside for loop write : summ=summ+i
- summ+=i or summ=summ+i
- dont give variable name as **sum**

In [ ]:

```
# summ=0
# forloop:
#   summ=summ+i
summ=0 # Intial point
for i in range(1,11):
    summ=summ+i  # updation

# step-1: summ=0  i=1   summ=0+1=1
# step-2: summ=1  i=2   summ=1+2=3
# step-3: summ=2  i=3   summ=3+3=6
```

In [13]:

```
summ=0 # Intial point
for i in range(1,11):
    summm=summm+i  # updation
    print(summm)
```

```
1
3
6
10
15
21
28
36
45
55
```

In [14]:

```
summm=0 # Intial point
for i in range(1,11):
    summm=summm+i  # updation

print(summm)
```

```
55
```

In [15]:

```
#Q8) Find the average of first 10 natural numbers
summm=0 # Intial point
for i in range(1,11):
    summm=summm+i  # updation

print(summm/10)
```

```
5.5
```

In [24]:

```
summm=0
def add():
    global summm,n1,n2
    n1=eval(input("enter the start:"))
    n2=eval(input("enter the last num:"))
    for i in range(n1,n2):
        summm=summm+i
    return(summm)
summm=add()
avg=summm/(n2-n1)
print(avg)
```

```
5.5
```

### Counter wrapper

In [25]:

```
for i in range(3):
    print("hello")
```

```
hello
hello
hello
```

- count=0
- for loop
- whenerver the creteria meet count=count+1

- in the summation we use sum=sum+i
- in the counter we use count=count+1

In [26]:

```
# q9)
count=0
for i in range(3):
    print("hello")
    count=count+1

print(count)
```

```
hello
hello
hello
3
```

In [28]:

```
# Q10)
count=0
num=eval(input("enter the number:"))
for i in range(1,num+1):
    if num%i==0:
        print(i)
        count=count+1
print("The number of divisions are:",count)
```

```
1
3
5
15
25
75
The number of divisions are: 6
```

In [29]:

```
# Q11)
count,summ=0,0
num=eval(input("enter the number:"))
for i in range(1,num+1):
    if num%i==0:
        print(i)
        count=count+1
        summ=summ+i
print("The number of divisions are:",count)
print("The summation of all divisions are:",summ)
```

```
1
3
5
15
25
75
The number of divisions are: 6
The summation of all divisions are: 124
```

In [ ]:

```
# Q12)
# ask the user get a random number: n1
# ask the user enter a number: n2
# if n1==n2: print(won)
# otherwise print(fals)
# i want to give 3 chances
```

In [35]:

```
import random
for i in range(3):
    num1 = random.randint(1,10)
    print(num1)
    num2 = eval(input("Enter your number"))
    if num1 == num2:
        print("Congrats yo've won")
        break
    else:
        print("Failed,mate")
```

```
3
Failed,mate
1
Congrats yo've won
```

In [ ]:

```
case-1: whever you won code should stop (completed)

case-2: whenever you fail
        it should display how many chances you have left

case-3: whenever all the chances are over
        it should display
        try after 24 hours
```

In [ ]:

```
yes sir didnt get prime number problem...pls explain
answer know or dontknow

how to write the code
if you know the answer === how to write the code
dont know
```

In [ ]:

```
n%1==0  and n%n==0

n%<>!=0
```

In [3]:

```
n=eval(input("Enter a number: "))
if n>1:
    for i in range(1,n):
        if n%i==0:
            print("The given number is not Prime Number")
        break
else:
    print("The given number is Prime Number.")

# n=7
# if 7>1  True ==== >
#         i (1,7)
#           7%1  7%2  7%3 7%4  7%5  7%6    where is 7%7 ?
```

```
The given number is not Prime Number
```

In [ ]:

```
# when to write loop
if num%1==0:
    print('1 is the divisor of num')

if num%2==0:
    print('2 is the divisor')
```

In [ ]:

```
num=eval(input('Enter any number : '))
for i in range(2,num):
    if num%i==0:
        print('Not prime')
        break
    else:
        print('Prime')
        break
```

In [ ]:

```
count=0
num=eval(input('enter number:'))
for i in range(num,num+1):
    if num%i==0:
        print(i,end=" ")
        count=count+1
        if count==2:
            print('it is prime number')
        else:
            print('it is not prime')
```

In [ ]:

```
- if 1 and 2 never be consider as prime number

- for loop here 3 to n
-      condition
```

In [ ]:

```
# Today also you try it
```

In [4]:

```
# wap ask the user take 5 random numbers
# provide me the count
# in that how many are even and how many are odd
import random
even_count=0
odd_count=0
for i in range(5):
    num=random.randint(1,100)
    if num%2==0:
        even_count+=1  # even_count=even_count+1
    else:
        odd_count+=1

print("The total number of evens are:",even_count)
print("The total number of odds are:",odd_count)
```

```
The total number of evens are: 2
The total number of odds are: 3
```

In [12]:

```
import random
even_sum=0
odd_sum=0
for i in range(5):
    num=random.randint(1,100)
    if num%2==0:
        print(f"{num} is an even")
        even_sum+=num  # even_count=even_count+1
    else:
        print(f"{num} is an odd")
        odd_sum+=num
print(even_sum)
print(odd_sum)
```

```
19 is an odd
47 is an odd
38 is an even
41 is an odd
93 is an odd
38
200
```

**in operator**

- For loop iteration possible with two ways
    - range
    - in
- range is math related, so it takes numbers
- in directly access with characters
- in the naive way in operator only for strings

In [15]:

```
str1='python'

'p' in str1 #  is 'p' available in str1

'y' in str1

'P' in str1
```

Out[15]:

```
False
```

In [ ]:

```
str1='python'

'p' in str1 #  is 'p' available in str1
'y' in str1
't' in str1
'h' in str1
'o' in str1
'n' in str1

i in str1
```

In [16]:

```
for i in str1:
    print(i)
```

```
p
y
t
h
o
n
```

In [17]:

```
for i in 'hello':
    print(i)
```

```
h
e
l
l
o
```

In [18]:

```
# error
for i in range('hello'):
    print(i)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 for i in range('hello'):
      2     print(i)

TypeError: 'str' object cannot be interpreted as an integer
```

- range and eval will not process strings english characters
- range is a data type
- eval is not a data type

In [ ]:

```
'a'<'A'   #F
'a'>'A'   # T
'p'>'y'   # F
```

### ASCII

- Amercican Standard Code for Information Interchange
- Every charcter symbol associated with a number
- 'a' : 97 'A': 65
- 'b' : 98 'B': 66

### ord-chr

- ord and chr are in-built functions
- ord will give the ascii number of charcter
- chr will give the charcter of a number

In [20]:

```
ord('A'), ord('a')
```

Out[20]:

```
(65, 97)
```

In [21]:

```
'a'>'A'   # 97>65
```

Out[21]:

```
True
```

In [23]:
```
chr(65),chr(97)
```

Out[23]:
```
('A', 'a')
```

**task-1**

- Print all ascii number of A to Z using for loop

In [ ]:
```
for i in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

In [ ]:
```
#Package name: string
#import string
#dir string
#find out the method which method will give capitalletters
```

In [24]:
```
import string
```

In [25]:
```
dir(string)
```

Out[25]:
```
['Formatter',
 'Template',
 '_ChainMap',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '_re',
 '_sentinel_dict',
 '_string',
 'ascii_letters',
 'ascii_lowercase',
 'ascii_uppercase',
 'capwords',
 'digits',
 'hexdigits',
 'octdigits',
 'printable',
 'punctuation',
 'whitespace']
```

In [26]:
```
string.ascii_letters
```

Out[26]:
```
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

In [27]:
```
string.ascii_lowercase
```

Out[27]:
```
'abcdefghijklmnopqrstuvwxyz'
```

In [28]:
```
string.ascii_uppercase
```

Out[28]:
```
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

```
In [31]: for i in string.ascii_lowercase:
             print(f"{i}:{ord(i)}",end=' ')
```

```
a:97 b:98 c:99 d:100 e:101 f:102 g:103 h:104 i:105 j:106 k:107 l:108 m:109 n:110
o:111 p:112 q:113 r:114 s:115 t:116 u:117 v:118 w:119 x:120 y:121 z:122
```

```
In [32]: for i in string.ascii_uppercase:
             print(f"{i}:{ord(i)}",end=' ')
```

```
A:65 B:66 C:67 D:68 E:69 F:70 G:71 H:72 I:73 J:74 K:75 L:76 M:77 N:78 O:79 P:80
Q:81 R:82 S:83 T:84 U:85 V:86 W:87 X:88 Y:89 Z:90
```

```
In [33]: for i in string.punctuation:
             print(f"{i}:{ord(i)}",end=' ')
```

```
!:33 ":34 #:35 $:36 %:37 &:38 ':39 (:40 ):41 *:42 +:43 ,:44 -:45 .:46 /:47 ::58
;:59 <:60 =:61 >:62 ?:63 @:64 [:91 \:92 ]:93 ^:94 _:95 `:96 {:123 |:124 }:125 ~:1
26
```

```
In [37]: for i in range(33,127):
             print(f"{i}:{chr(i)}",end=' ')
```

```
33:! 34:" 35:# 36:$ 37:% 38:& 39:' 40:( 41:) 42:* 43:+ 44:, 45:- 46:. 47:/ 48:0 4
9:1 50:2 51:3 52:4 53:5 54:6 55:7 56:8 57:9 58:: 59:; 60:< 61:= 62:> 63:? 64:@ 6
5:A 66:B 67:C 68:D 69:E 70:F 71:G 72:H 73:I 74:J 75:K 76:L 77:M 78:N 79:O 80:P 8
1:Q 82:R 83:S 84:T 85:U 86:V 87:W 88:X 89:Y 90:Z 91:[ 92:\ 93:] 94:^ 95:_ 96:` 9
7:a 98:b 99:c 100:d 101:e 102:f 103:g 104:h 105:i 106:j 107:k 108:l 109:m 110:n 1
11:o 112:p 113:q 114:r 115:s 116:t 117:u 118:v 119:w 120:x 121:y 122:z 123:{ 124:
| 125:} 126:~
```

```
In [42]: for i in range(9758,10000):
             print(f"{i}:{chr(i)}",end=' ')
```

```
9758:☞ 9759:☟ 9760:☠ 9761:☡ 9762:☢ 9763:☣ 9764:☤ 9765:☥ 9766:☦ 9767:☧ 9768:☨ 9
769:☩ 9770:☪ 9771:☫ 9772:☬ 9773:☭ 9774:☮ 9775:☯ 9776:☰ 9777:☱ 9778:☲ 9779:☳
9780:☴ 9781:☵ 9782:☶ 9783:☷ 9784:☸ 9785:☹ 9786:☺ 9787:☻ 9788:☼ 9789:☽ 9790:☾
9791:☿ 9792:♀ 9793:♁ 9794:♂ 9795:♃ 9796:♄ 9797:♅ 9798:♆ 9799:♇ 9800:♈ 9801:♉
9802:♊ 9803:♋ 9804:♌ 9805:♍ 9806:♎ 9807:♏ 9808:♐ 9809:♑ 9810:♒ 981
1:♓ 9812:♔ 9813:♕ 9814:♖ 9815:♗ 9816:♘ 9817:♙ 9818:♚ 9819:♛ 9820:♜ 9821:
♝ 9822:♞ 9823:♟ 9824:♠ 9825:♡ 9826:♢ 9827:♣ 9828:♤ 9829:♥ 9830:♦ 9831:♧ 9832:
♨ 9833:♩ 9834:♪ 9835:♫ 9836:♬ 9837:♭ 9838:♮ 9839:♯ 9840:♰ 9841:♱ 9842:♲ 9843:♳
9844:♴ 9845:♵ 9846:♶ 9847:♷ 9848:♸ 9849:♹ 9850:♺ 9851:♻ 9852:♼ 9853:♽ 9854:
♾ 9855:♿ 9856:⚀ 9857:⚁ 9858:⚂ 9859:⚃ 9860:⚄ 9861:⚅ 9862:⚆ 9863:⚇ 9864:⚈ 986
5:⚉ 9866:⚊ 9867:⚋ 9868:⚌ 9869:⚍ 9870:⚎ 9871:⚏ 9872:⚐ 9873:⚑ 9874:⚒ 9875:⚓
9876:⚔ 9877:⚕ 9878:⚖ 9879:⚗ 9880:⚘ 9881:⚙ 9882:⚚ 9883:⚛ 9884:⚜ 9885:⚝ 9886:⚞
9887:⚟ 9888:⚠ 9889:⚡ 9890:⚢ 9891:⚣ 9892:⚤ 9893:⚥ 9894:⚦ 9895:⚧ 9896:⚨ 9897:
⚩ 9898:⚪ 9899:⚫ 9900:⚬ 9901:⚭ 9902:⚮ 9903:⚯ 9904:⚰ 9905:⚱ 9906:⚲ 9907:⚳ 990
8:⚴ 9909:⚵ 9910:⚶ 9911:⚷ 9912:⚸ 9913:⚹ 9914:⚺ 9915:⚻ 9916:⚼ 9917:⚽ 9918:⚾ 99
19:⚿ 9920:⛀ 9921:⛁ 9922:⛂ 9923:⛃ 9924:⛄ 9925:⛅ 9926:⛆ 9927:⛇ 9928:⛈ 992
9:⛉ 9930:⛊ 9931:⛋ 9932:⛌ 9933:⛍ 9934:⛎ 9935:⛏ 9936:⛐ 9937:⛑ 9938:⛒ 9939:
⛓ 9940:⛔ 9941:⛕ 9942:⛖ 9943:⛗ 9944:⛘ 9945:⛙ 9946:⛚ 9947:⛛ 9948:⛜ 9949:⛝
9950:⛞ 9951:⛟ 9952:⛠ 9953:⛡ 9954:⛢ 9955:⛣ 9956:⛤ 9957:⛥ 9958:⛦ 9959:⛧ 996
0:⛨ 9961:⛩ 9962:⛪ 9963:⛫ 9964:⛬ 9965:⛭ 9966:⛮ 9967:⛯ 9968:⛰ 9969:⛱ 9970:
⛲ 9971:⛳ 9972:⛴ 9973:⛵ 9974:⛶ 9975:⛷ 9976:⛸ 9977:⛹ 9978:⛺ 9979:⛻ 998
0:⛼ 9981:⛽ 9982:⛾ 9983:⛿ 9984:✀ 9985:✁ 9986:✂ 9987:✃ 9988:✄ 9989:✅ 999
0:✆ 9991:✇ 9992:✈ 9993:✉ 9994:✊ 9995:✋ 9996:✌ 9997:✍ 9998:✎ 9999:✏
```

```
In [ ]: 
```