Loops

- reuse the code for multiple iterations unless the given condition is satifies
- we have 2 loops
  - for loop
  - while loop
- any loop has 3 types
  - start point
  - condition to stop
  - increment or decrement
- In for loop we will apply all above three in a single line
- But in while loop we will apply all abive three in 3 different lines,but 3 things are same

*For loop**

method 1

syntax;

for i in range(): print(i)

In [1]:
```
for i in range(3):
    print(i)
```

```
0
1
2
```

note

- the default start point is 0
- the default increment value is 1
- last value=stop-1

range(5)

- start value=0
- increment value is 1
- last value is 4

In [6]:
```
for i in range(5):
    print(i,end=' ')
```

```
0 1 2 3 4
```

In [10]:
```
for i in range(3):
    print("hello")
```

```
hello
hello
hello
```

In [11]:
```
for i in range(3):
    print("hello",end=' ')
```

```
hello hello hello
```

In [13]:
```
for i in range(3):
    print(i,"hello")
```

```
0 hello
1 hello
2 hello
```

** method-2**

**range(start,end)

- start value=start
- default increment by 1
- last value =stop-1

In [ ]:
```
#syntax

for i in range(<start>,<stop>):
    print(i)
```

In [15]:
```
#example:

for i in range(2,6):
    print(i)
```

```
2
3
4
5
```

In [20]:
```
#wap ask the user  print the square of numbers from 1to 5
for i in range(1,6):
    print(i*i)
```

```
1
4
9
16
25
```

In [28]:
```
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 [30]:
```python
#wap ask the user take 5 random number and print square

import random
for i in range(5):
    num=random.randint(10,100)
    print(f"the square of {num}  is {num*num}")
```

```
the square of 67  is 4489
the square of 28  is 784
the square of 83  is 6889
the square of 95  is 9025
the square of 91  is 8281
```

In [9]:
```python
#create a function a resued the call that function call the function  inside the

import random
def square():
    n=random.randint(10,100)
    print(f"the square of {n}  is {n*n}")
for i in range(5):
     square()
```

```
the square of 41  is 1681
the square of 29  is 841
the square of 63  is 3969
the square of 15  is 225
the square of 94  is 8836
```

In [50]:
```python
#wap ask the user  print the even and odd values between 11 and 21

for i in range(11,21):
    if i%2==0:
        print(f"{i} even number")
    else:
        print(f"{i}odd number")
```

```
11odd number
12 even number
13odd number
14 even number
15odd number
16 even number
17odd number
18 even number
19odd number
20 even number
```

In [3]:
```python
#wap a program ask the user enter a value 5 times print that value even or odd
for i in range(5):
    num=eval(input("enter a number:"))
    if num%2==0:
        print(f"{num} is even")
    else:
        print(f"{num} is odd")
```

```
4 is even
7 is odd
8 is even
9 is odd
8 is even
```

method-3

range(start,stop,step)

- start:always a start value
  - if increment value is positive
    - positive value means positive directions(Forward)
      - last=stop-1
  - if increment value is negative
  - negative value means negative directions(backward)
  - last=stop+1

In [54]:
```
#case-1
for i in range(2,20,2):
    print(i,end=' ')

# start value=2
#stop value=20
#step-2  +ve direction

#first write start value and stop value
#Then look at direction positive or negative
#then make the conclusion
```

```
2 4 6 8 10 12 14 16 18
```

In [2]:
```
#case-2:                          #no results
for i in range(2,20,-2):
    print(i,end=" ")

#start value=2
#stop value=20
#step=-2 negative
#last value=stop+1===>20+1=21
```

In [3]:
```
#case-3:

for i in range(2,-20,-2):
    print(i,end=" ")

#start value=2
#stop value=-20
#step=2 negatuve
#last value=stop+1===>-20+1=-19
```

```
2 0 -2 -4 -6 -8 -10 -12 -14 -16 -18
```

In [7]:
```
#case-4:

for i in range(-2,-20,-2):
    print(i,end=" ")
```

```
#start value=-2
#step value=-2 negative
#last value=stop+1===>-20+1=-19
```

```
-2 -4 -6 -8 -10 -12 -14 -16 -18
```

In [8]:
```
#case-5                                        #not possible

for i in range(-2,20,-2):
    print(i,end=" ")
#start value=-2
#step value=-2 negative
#last=stop+1===>20+1=21
```

In [2]:
```
#wap to create 9th table

for i in range(1,11,1):
    print(f"9x{i}={9*i}")
```

```
9x1=9
9x2=18
9x3=27
9x4=36
9x5=45
9x6=54
9x7=63
9x8=72
9x9=81
9x10=90
```

In [28]:
```
#wap to create 50 divisors

for i in range(1,51,1):
    if 50%i==0:
        print(f"{i} ")
```

```
1
2
5
10
25
50
```

In [36]:
```
def divisors():
    n=eval(input("enter which number divisors you want"))
for i in range(1,n):
    if n%i==0:
        print(f"{i}")
divisors()
```

```
1
3
5
15
25
```

In [50]:
```
def divisors(num):

    for i in range(1,num+1):
        if num%i==0:
            print(f"{i}")
```

```
divisors(90)
```

```
1
2
3
5
6
9
10
15
18
30
45
90
```

In [66]:
```
#wap to print 1st natural numbers sum
sum=0
for i in range(1,11):
    sum=sum+i
print(f"The sum of first 10 natural numbers are {sum}")
```

```
The sum of first 10 natural numbers are 55
```

In [76]:
```
import keyword

keyword.kwlist
```

Out[76]:
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

counter wapper

In [83]:
```
#wap how many divisors available for 50

count=0
for i in range(1,51):
    if 50%i==0:
        count=count+1

print(f"the number of divisors are{count}")
```

```
the number of divisors are6
```

In [89]:
```
#wap ask the user enter the divisor how many divisor are there
count=0
num=eval(input("Enter a number which want divisor"))
for i in range(1,num+1):
    if num%i==0:
        count=count+1

print(f"the number of divisors are {count}")
```

```
the number of divisors are 4
```

In [98]:
```
#ask the user enter number1 from keyboard
#ask the random number2 from(1-10)
import random


for i in range(3):
    num1=eval(input("Enter a nuumber1: "))
    num2=random.randint(1,10)
    if num1==num2:
        print("you won")
    else:
        print("you lost")
```

```
you lost
you lost
you lost
```

In [101…
```
import random


for i in range(3):
    num1=random.randint(1,10)
    print(num1)
    num2=eval(input("enter a number"))
    if num1==num2:
        print("you won")
        break
    else:
        print("you lost")
```

```
9
you won
```

In [22]:
```
#case-2:give five chances
# whenever fails print the number of chances left

#case-3:
#suppose your all chances are over your account is block for 24 hrs

import random

for i in range(5):
    num1=random.randint(1,50)
    num2=eval(input("Enter a number"))
    count=4
    if num1==num2:
        print("you won")
        break
    else :
        print("you loss")
        print(f"chances left are {count-i}")
if count-i==0:
    print("your account is blocked for 24 hours")
    print("please try again 24 hours")
```

```
you loss
chances left are 4
you loss
chances left are 3
```

```
you loss
chances left are 2
you loss
chances left are 1
you loss
chances left are 0
your account is blocked for 24 hours
please try again 24 hours
```

** in operator**

- there are 2 operators are in for loop
  - one is **range** operator
  - second one is: **in** operator
  - range operator mainly for numbers for i in range
  - **in** operator for characters

In [2]:
```
name="python"
'p' in name    #True (p is included in python)
'y' in name     #True  (y is included in python)
's' in name     #False (s is not included in python)
```

Out[2]: False

In [3]:
```
#syntax
for i in name:
    print(i)

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

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

In [4]:
```
for i in name:
    print(i,end= '')
```

```
python
```

In [5]:
```
for i in 'name':
    print(i,end= '')
```

```
name
```

In [6]:
```
'A'=='a'    #ascii values A value=65 and a=97
```

Out[6]: False

In [7]:
```
'A'>'a'    #depend on ascii values[american standard code for information interc
```

Out[7]: False

ord-chr

- ord keyword gives ascii values
- char keyword which provides characters for given ASCII values

In [10]:
```
ord('A')
```

Out[10]: 65

In [11]:
```
ord('A'),ord('a')
```

Out[11]: (65, 97)

In [12]:
```
chr(65),chr(97)
```

Out[12]: ('A', 'a')

In [ ]:
```
# i want print all ascii values using for loop A to Z
ord('A')
ord('B')
ord('C')
----
----
----
ord('Z')

ord(i)
```

In [28]:
```
for i in string.ascii_uppercase:
        print(f"The ascii value of {i} is {ord(i)}")
```

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

package name Strings

In [15]:
```
import string
dir(string)
```

Out[15]:
```
['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 [22]:
```
string.ascii_uppercase
```

Out[22]: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

In [24]:
```
string.ascii_lowercase
```

Out[24]: 'abcdefghijklmnopqrstuvwxyz'

In [34]:
```
#I want to know how many ascii values are there available

for i in range(1,150):
    print(f"{i},{chr(i)}",end=' ')
```

```
1,┌ 2,┐ 3,└ 4,┘ 5,│ 6,─ 7,● 8 9,          10,
 14,♫ 15,☼ 16,┼ 17,◄ 18,↕ 19,‼ 20,¶ 21,┴ 22,┬ 23,┤ 24,↑ 25,├ 26,→ 27,← 28,∟ 29,↔
30,▲ 31,▼ 32,  33,! 34," 35,# 36,$ 37,% 38,& 39,' 40,( 41,) 42,* 43,+ 44,, 45,- 4
6,. 47,/ 48,0 49,1 50,2 51,3 52,4 53,5 54,6 55,7 56,8 57,9 58,: 59,; 60,< 61,= 6
2,> 63,? 64,@ 65,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 7
8,N 79,O 80,P 81,Q 82,R 83,S 84,T 85,U 86,V 87,W 88,X 89,Y 90,Z 91,[ 92,\ 93,] 9
4,^ 95,_ 96,` 97,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 111,o 112,p 113,q 114,r 115,s 116,t 117,u 118,v 119,w 120,x 121,y 1
22,z 123,{ 124,| 125,} 126,~ 127,□ 128,□ 129,□ 130,□ 131,□ 132,□ 133,□ 134,□ 13
5,□ 136,□ 137,□ 138,□ 139,□ 140,□ 141,□ 142,□ 143,□ 144,□ 145,□ 146,□ 147,□ 148,□
149,□
```

In [ ]:

In [ ]:

In [ ]: