In [49]:
```
if True: # indentiation is always 4 spaces
    print('Data Science')
```

```
Data Science
```

In [50]:
```
if False:
    print('Data Science')
print('bye for now')
```

```
bye for now
```

In [51]:
```
if True: # indentiation is always 4 spaces
    print('Data Science')
print('bye for now')
```

```
Data Science
bye for now
```

Lets do one program as if number is divide by 2 then reminder is 0 then it is even number if reminder is not 0 then it is odd number

In [52]:
```
#to print only even number

x = 14
r = x % 2

if r == 0:
    print('Even number')

if r == 1:
    print('Odd Number')
```

```
Even number
```

In [53]:
```
#to print only even number

x = 100
r = x % 2

if r == 0:
    print('Even number')
else:
    print('Odd Number')
```

```
Even number
```

In [54]:
```
#to print only even number
x = 11
r = x % 2

if r == 0:
    print('Even number')
```

In [55]:
```
x = 5
r = x % 2

if r == 0:
    print('Even number')
print('odd number')
```

```
odd number
```

In [56]:
```
x = 8
r = x % 2

if r == 0:
    print('Even number')
print('odd number')
```

```
Even number
odd number
```

In [57]:
```
x = 8
r = x % 2

if r == 0:
    print('Even number')
if r == 1:
    print('odd number')
```

```
Even number
```

In [58]:
```
x = 7
r = x % 2

if r == 0:
    print('Even number')
if r == 1:
    print('odd number')
```

```
odd number
```

In [59]:
```
x = 13
r = x % 2

if r == 0:
    print('Even number')

if r != 0:
    print('odd number')
```

```
odd number
```

In [60]:
```
x = 56
r = x%2

if r==0:
    print('even number')

if r!=0:
    print('odd number')
```

```
even number
```

if we observe the code its too many line cuz many of the coder always they wanted to reduce the code lenght which is very good practise. instead of 2 if we can use if-- else

In [61]:
```
x = 10
r = x%2

if r == 0:
```

```
    print('even number')
else:
    print('odd number')
```

```
even number
```

In [62]:
```
x = 13
r = x % 2

if r == 0:
    print('even number')

else:
    print('odd number')
```

```
odd number
```

In [63]:
```
x = 10
r = x % 2

if r == 0:
    print('even number')
    if r>=5:
        print('greater number')

else:
    print('odd number')
```

```
even number
```

In [64]:
```
x = 11
r = x % 2

if r == 0:
    print('even number')
    if r>=5:
        print('greater number')

else:
    print('odd number')
```

```
odd number
```

In [65]:
```
x = 10
r = x % 2

if r == 0:
    print('even number')
    if r>=0:
        print('greater number')


else:
    print('odd number')
```

```
even number
greater number
```

NESTED IF (if we have 2 condition so we need to implment with nested if)

In [66]:
```
x = 6
r = x % 2
```

```
if r == 0:
    print('Even number')
    if x>5:
        print('greater number')
    else:
        print('not greater')
else:
    print('Odd Number')
```

```
Even number
greater number
```

In [67]:
```
x = 15
r = x % 2
if r == 0:
    print('even number')
    if x>5:
        print('greater number')
    else:
        print('not greater')

else:
    print('odd number')
```

```
odd number
```

In [68]:
```
x = 2
r = x % 2

if r == 0:
    print('Even number')
    if x>5:
        print('greater number')
    else:
        print('not greater')
else:
    print('Odd Number')
```

```
Even number
not greater
```

We do have concept of (IF - ELIF- ELSE) e.g i want to print (1--> one , 2 --> two, 3--> three, 4--> four, 5- five)

In [69]:
```
x = 1

if x == 1:
    print('one')

if x == 2:
    print('two')

if x == 3:
    print('three')

if x == 4:
    print('four')
```

```
one
```

In [70]:
```
x = 2

if(x == 1):
    print('one')
elif(x == 2):
    print('Two')
elif(x == 3):
    print('Three')
elif(x == 4):
    print('four')
```

```
Two
```

In [71]:
```
x = 3

if(x == 1):
    print('one')
elif(x == 2):
    print('Two')
elif(x == 3):
    print('Three')
elif(x == 4):
    print('four')
```

```
Three
```

In [72]:
```
x = 5

if(x == 1):
    print('one')
elif(x == 2):
    print('Two')
elif(x == 3):
    print('Three')
elif(x == 4):
    print('four')
else:
    print('wrong output')
```

```
wrong output
```

In [73]:
```
print('data science')
```

```
data science
```

In [74]:
```
print('data science')
print('data science')
```

```
data science
data science
```

LOOPS -- in programing world some time we keep on repeating , may be you want to repeat 5 statement so one way is copy & paste multiple times or other way is. if you want to print the datascience 10 times then what you will you cant copy for 10 times , if you want to print 1000 times then you cant do manualy . that is the reason why we need to apply loop -> 2 type of loops -- While loop & For loop

In [75]:
```
i = 1
while i<=7:
```

```
    print('data science')
    i = i+1
```

```
data science
data science
data science
data science
data science
data science
data science
```

In [76]:
```
i = 5          # initializing
while i>=1:    # condition
    print('data science')
    i = i - 1  # decrement
```

```
data science
data science
data science
data science
data science
```

In [77]:
```
i = 1
while i <= 6:
    print('data science',':',i)
    i = i+1
```

```
data science : 1
data science : 2
data science : 3
data science : 4
data science : 5
data science : 6
```

In [78]:
```
i = 6
while i >=1:
    print('data science',':',i)
    i=i-1
```

```
data science : 6
data science : 5
data science : 4
data science : 3
data science : 2
data science : 1
```

can we use multiple while loop || nested while loop

In [79]:
```
i = 1

while i<=5:
    print(' data science') # when we mention end then new line will not create
    j = 1
    while j<=4:
        print(' technology')
        j = j + 1

    i = i + 1
    print()
```

```
data science
technology
technology
technology
technology

data science
technology
technology
technology
technology

data science
technology
technology
technology
technology

data science
technology
technology
technology
technology

data science
technology
technology
technology
technology
```

In [80]:
```
i = 1
while i<=5:
    print(' data science', end = "") # when we mention end then new line will no
    j = 1
    while j<=4:
        print(' technology', end="")
        j = j + 1

    i = i + 1
    print()
```

```
data science technology technology technology technology
data science technology technology technology technology
data science technology technology technology technology
data science technology technology technology technology
data science technology technology technology technology
```

In [81]:
```
i = 1
while i<=5:
    print(' data science', end = "  *") # when we mention end then new line will
    j = 1
    while j<=4:
        print(' technology', end="  *")
        j = j + 1

    i = i + 1
    print()
```

```
data science  * technology  * technology  * technology  * technology  *
data science  * technology  * technology  * technology  * technology  *
data science  * technology  * technology  * technology  * technology  *
data science  * technology  * technology  * technology  * technology  *
data science  * technology  * technology  * technology  * technology  *
```

In [82]:
```
i = 1
while i <= 4 :
    j = 0
    while  j <= 3 :
        print(i*j, end=" ")
        j += 1
    print()
    i += 1
```

```
0 1 2 3
0 2 4 6
0 3 6 9
0 4 8 12
```

FOR LOOP - normally while loop it work with condition but for loop it will work with sequence (list, string,int)

In [83]:
```
name = 'nit'
for i in name:
    print(i)
```

```
n
i
t
```

In [84]:
```
name1 = [1,3.5,'hallo']

for i in name1:
    print(i)
```

```
1
3.5
hallo
```

In [85]:
```
for i in [2, 3, 7.8, 'hi']:
    print(i)
```

```
2
3
7.8
hi
```

In [86]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [87]:
```
for i in range(1,5):
    print(i)
```

```
1
2
3
4
```

In [88]:
```
for i in range(1,10,3):
    print(i)
```

```
1
4
7
```

In [89]:
```
# print the numer which is not divisible by 5

for i in range(1,11):

    if i%5 != 0 :
      print(i)
```

```
1
2
3
4
6
7
8
9
```

LETS DISCUSS ABOUT 3 KEYWORDS -- BREAK || CONTINUE || PASS

BREAK STATEMNT - if you apply break statment in a loop then it will end the loop

Pass = skips block of code(function, class etc)

Continue= skips 1 step/iteration during loop

Break= jumps out of the function/loop

In [90]:
```
for i in range(1,11):
    print(i)
```

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

In [91]:
```
for i in range(1,11):
    if i == 5:
        break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
```

In [92]:
```
for i in range(1,11):
    if i == 5:
        break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
    print(i)
```

```
1
2
3
4
```

In [93]:
```
for i in range(1,11):
    if i == 5:
        break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
print(i)
```

```
5
```

In [94]:
```
# in continue ,  loop wont be terminate



for i in range(1,11):
    if i == 5:
        continue
    print(i)
```

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

In [95]:
```
for i in range(1,11):
    if i == 5:
        continue
    print('hello ',i)
```

```
hello  1
hello  2
hello  3
hello  4
hello  6
hello  7
hello  8
hello  9
hello  10
```

In [96]:
```
#PASS Statement - pass the code & it wont go

for i in range(1,11):
```

```
  Cell In[96], line 3
    for i in range(1,11):
                         ^
SyntaxError: incomplete input
```

In [97]:
```
for i in range(1,11):
    pass
```

In [98]:
```
# PRINTING PATTERN IN PYTHON
```

In [99]:
```
print('# # # #')
print('# # # #')
print('# # # #')
print('# # # #')
```

```
# # # #
# # # #
# # # #
# # # #
```

In [100…
```
for i in range(4):
    print('# # # #', )
```

```
# # # #
# # # #
# # # #
# # # #
```

In [101…
```
for j in range(4):
    print('#', end="  ")
```

```
#  #  #  #
```

In [102…
```
for j in range(4):
    print('#', end="  ")

for j in range(4):
    print('#', end="  ")
```

```
#  #  #  #  #  #  #  #
```

In [103…
```
for j in range(4):
    print('#', end="  ")

print()
```

```
for j in range(4):
    print('#', end="  ")
```

```
#  #  #  #
#  #  #  #
```

for j in range(4): print('#', end=" ") print() for j in range(4): print('#', end=" ") print() for j in range(4): print('#', end=" ") print() for j in range(4): print('#', end=" ")

In [104…
```
for i in range(4):
    for j in range(4):
        print('#', end="  ")
    print()
    # pease use debug mode
```

```
#  #  #  #
#  #  #  #
#  #  #  #
#  #  #  #
```

In [105…
```
for i in range(5):
    for j in range(i):
        print('#', end="  ")
    print()
```

```
#
#  #
#  #  #
#  #  #  #
```

In [106…
```
for i in range(4):
    for j in range(i+1):
        print('#', end="  ")
    print()
```

```
#
#  #
#  #  #
#  #  #  #
```

In [107…
```
for i in range(4):
    for j in range(4-i):
        print('*', end="  ")
    print()
```

```
*  *  *  *
*  *  *
*  *
*
```

In [108…
```
for i in range(4):
    for j in range(4-i):
        print('#', end="  ")
    print()
```

```
#  #  #  #
#  #  #
#  #
#
```

For|Else in python

in other language for else not supportable but in python it is supportable eg- lets print the number from 1- 20 & we dont want print number which is divisible by 5

In [109…
```
nums = [12,15,18,21,26]

for num in nums:
    if num % 5 == 0:
        print(num)
```

```
15
```

In [110…
```
nums = [12,14,18,21,25]
for num in nums:
    if num % 5 == 0:
        print(num)
```

```
25
```

In [111…
```
nums = [12,14,18,21,25,20]
for num in nums:
    if num % 5 == 0:
        print(num)
```

```
25
20
```

In [112…
```
nums = [12,14,18,21,25,20]
for num in nums:
    if num % 5 == 0:
        print(num)
        break
```

```
25
```

In [113…
```
nums = [10,14,18,21,20,25]
for num in nums:
    if num % 5 == 0:
        print(num)
        break #it will print only 1 number then it break
```

```
10
```

In [114…
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we go
for num in nums:
    if num % 5 == 0:
        print(num)
       # break
```

In [115…
```
nums = [7,14,18,21,23,27,22] #hear there is no number which is divisible by 5 we
for num in nums:
    if num % 5 == 0:
        print(num)
        break
    else:
        print('Number Not Found') #every iteration it cheking condition
```

```
Number Not Found
Number Not Found
Number Not Found
Number Not Found
Number Not Found
Number Not Found
Number Not Found
```

In [116…
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we go
for num in nums:
    if num % 5 == 0:
        print(num)
        #break
else:
        print('Not Found') # hear else we dont write in if block but we can writ
```

```
Not Found
```

In [117…
```
nums = [10,14,18,21,20,27] #hear there is no number which is divisible by 5 we g
for num in nums:
    if num % 5 == 0:
        print(num)
        #break
else:
        print('Not Found')
```

```
10
20
Not Found
```

In [118…
```
nums = [10,14,18,21,20,27] #hear there is no number which is divisible by 5 we g
for num in nums:
    if num % 5 == 0:
        print(num)
        break
else:
        print('Not Found')
```

```
10
```

prime number - how to check given number is prime number or not

In [119…
```
num = 12

for i in range(2,num):
    if num % i == 0:
        print('Not prime Number')
        break
else:
    print('Prime Number')
```

```
Not prime Number
```

In [120…
```
num = 13

for i in range(2,num):
    if num % i == 0:
        print('Not prime Number')
```

```
        break
else:
    print('Prime Number')
```

```
Prime Number
```

In [ ]:
```
from array import *

arr = array('i',[])

n = int(input('Enter the length of the array'))

for i in range(5):
    x = int(input('Enter the next value'))
    arr.append(x)
    print(arr)
```

In [ ]:
```
# Way of creating array using numpy
```

In [ ]:
```
from numpy import *
arr = array([1,2,3,4,5])
print(arr)
type(arr)
```

In [ ]:
```
print(arr.dtype)
```

In [ ]:
```
arr = array([1,2,3,4,5.9])
print(arr)

print(arr.dtype)
```

In [ ]:
```
arr2 = array([1,2,3,4,5.9],float)
arr2
```

In [ ]:
```
import numpy as np
```

In [ ]:
```
arr4 = np.linspace(0, 16, 10) # break the code between 10 spaces between 0 to 16
arr4
```

In [ ]:
```
arr5 = np.arange(0,10,2) # arange - as range
arr5
```

In [ ]:
```
arr6 = np.zeros(5)
arr6
```

In [ ]:
```
arr7 = np.ones(5)
arr7
```

In [ ]:
```

```