special keywords

if if else if elif else nested if

In [2]:
```
if True:   #indentiation is always 4 spaces
    print('Ayushi patil')
```

```
Ayushi patil
```

In [3]:
```
if False:
    print('Ayushi Patil')
```

In [4]:
```
if False:
    print('Ayushi Patil')
print('bye for now')
```

```
bye for now
```

In [5]:
```
if True:
    print('Ayushi Patil')
print('bye for now')
```

```
Ayushi Patil
bye for now
```

In [6]:
```
# To print only even number

x=4
r=x%2

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

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

```
Even number
```

In [7]:
```
#to print only even number
x=11
r=x%2

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

In [8]:
```
x=5
r=x%2

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

```
odd number
```

In [9]:
```
x = 8
r=x%2

if   r==0:
```

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

```
even number
odd number
```

In [10]:
```
x=8
r=x%2

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

```
even number
```

In [11]:
```
x=7
r=x%2

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

```
odd number
```

In [12]:
```
x=13
r=x%2

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

```
odd 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 [13]:
```
x=2
r=x%2

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

```
even number
```

In [14]:
```
x=4
r=x%2

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

```
even number
```

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

In [15]:
```
x=6
r=x%2

if   r==0:
```

```
   print('even number')
   if x>5:
        print('greater number')
   else:
        print( 'not greated')
else:
   print('odd number')
```

```
even number
greater number
```

In [16]:
```
x=2
r=x%2

if   r==0:
   print('even number')
   if x>5:
        print('greater number')
   else:
        print( 'not greated')
else:
   print('odd number')
```

```
even number
not greated
```

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 [17]:
```
#when you use if it will check all condition but if we mention as elif then it w
x = 1
if(x==1):
    print('one')
if(x==2):
    print('two')
if(x==3):
    print('three')
if(x==4):
    print('four')
```

```
one
```

In [18]:
```
x = 2
if(x==1):
    print('one')
elif(x==2):
    print('two')
elif(x==3):
    print('three')
elif(x==4):
    print('four')
```

```
two
```

In [19]:
```
x = 5
if(x==1):
    print('one')
elif(x==2):
    print('two')
elif(x==3):
    print('three')
elif(x==4):
    print('four')
```

In [20]:
```
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 outout')
```

```
wrong outout
```

In [21]:
```
x = 15
if(x==1):
    print('one')
elif(x==2):
    print('two')
elif(x==3):
    print('three')
elif(x==4):
    print('four')
else:
    print('wrong outout')
```

```
wrong outout
```

In [22]:
```
print('data science')
```

```
data science
```

In [23]:
```
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 [24]:
```
i=1      #initializing
while i<=5:   # condition
    print('Datascience')
    i=i+1    #increment
```

```
Datascience
Datascience
Datascience
Datascience
Datascience
```

In [25]:
```
i=5
while i>=1:
    print('Datascience')
    i=i-1
```

```
Datascience
Datascience
Datascience
Datascience
Datascience
```

In [26]:
```
i=5           #initializing
while i>=1:   #condition
    print('Datascience',':',i)
    i=i-1     #decrement
```

```
Datascience : 5
Datascience : 4
Datascience : 3
Datascience : 2
Datascience : 1
```

can we use multiple while loop || nested while loop to understand nested whild indepth understand you can use pycharm debug with f8 option

In [27]:
```
i=1      #initializing
while i<=5:   # condition
    print('Datascience')  # when we mention end then new line will not create
    j=1
    while j<=4:
        print('technology')
        j=j+1
    i=i+1
    print()
```

```
Datascience
technology
technology
technology
technology

Datascience
technology
technology
technology
technology

Datascience
technology
technology
technology
technology

Datascience
technology
technology
technology
technology

Datascience
technology
technology
technology
technology
```

In [28]:
```
i=1      #initializing
while i<=5:   # condition
    print('Datascience',end="") # when we mention end then new line will not cre
    j=1
    while j<=4:
        print('technology',end="")
        j=j + 1

    i=i+1
    print()
```

```
Datasciencetechnologytechnologytechnologytechnology
Datasciencetechnologytechnologytechnologytechnology
Datasciencetechnologytechnologytechnologytechnology
Datasciencetechnologytechnologytechnologytechnology
Datasciencetechnologytechnologytechnologytechnology
```

In [29]:
```
i=1      #initializing
while i<=5:   # condition
    print('Datascience',end=' * ') # when we mention end then new line will not
    j=1
    while j<=4:
        print('technology',end=' * ')
        j=j+1
    i=i+1
    print()
```

```
Datascience * technology * technology * technology * technology *
Datascience * technology * technology * technology * technology *
Datascience * technology * technology * technology * technology *
Datascience * technology * technology * technology * technology *
Datascience * technology * technology * technology * technology *
```

In [30]:
```
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 [31]:
```
name='nit'
for i in name:
    print(i)
```

```
n
i
t
```

In [32]:
```
name1=[1,3.5,'hello']
for i in name1:
    print(i)
```

```
1
3.5
hello
```

In [33]:
```
for i in [2,3,7.8,'hi']:
    print(i)
```

```
2
3
7.8
hi
```

In [34]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [35]:
```
for i in range(1,5):
    print(i)
```

```
1
2
3
4
```

In [36]:
```
for i in range(1,10,3):
    print(i)
```

```
1
4
7
```

In [37]:
```
# 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
```

In [38]:
```
# can you write the python code for 5 multiplication table

for i in range(1,51):

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

```
5
10
15
20
25
30
35
40
45
50
```

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 [39]:
```
for i in range(1,11):
    print(i)
```

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

In [40]:
```
for i in range(1,11):
    if i == 5:
        break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
```

In [41]:
```
for i in range(1,11):
    if i == 5:
        break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
    print(i)
```

```
1
2
3
4
```

In [42]:
```
for i in range(1,11):
    if i == 5:
        break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
print(i)
```

```
5
```

in continue , loop wont be terminate

In [43]:
```
for i in range(1,11):
    if i == 5:
        continue
    print(i)
```

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

In [44]:
```
for i in range(1,11):
    if i == 5:
        continue
print(i)
```

```
10
```

In [45]:
```
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
```

PASS Statement - pass the code & it wont go

In [46]:
```
for i in range(1,11):
```

```
  Cell In[46], line 1
    for i in range(1,11):
                         ^
SyntaxError: incomplete input
```

In [67]:
```
for i in range(1,11):
    pass
```

PRINTING PATTERN IN PYTHON

In [69]:
```
print('# # # #')
print('# # # #')
print('# # # #')
print('# # # #')
```

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

In [71]:
```
for j in range(4):
    print('#',end=" ")
```

```
# # # #
```

In [73]:
```
for j in range(4):
    print('#',end=" ")
```

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

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

In [75]:
```
for j in range(4):
    print('#',end=" ")

print()

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

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

In [77]:
```
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 [79]:
```
for i in range(4):
    for j in range(4):
         print('#',end=" ")
    print()
     # pease use debug mode
```

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

In [81]:
```
for i in range(5):
    for j in range(i):
         print('#',end=" ")
    print()
```

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

In [83]:
```
for i in range(4):
    for j in range(i+1):
```

```
        print('#',end=" ")
    print()
```

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

In [85]:
```
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 [88]:
```
nums=[12,15,18,21,26]

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

```
15
```

In [90]:
```
for i in range(1,11):
    if i % 5 == 0:
        print(i)
```

```
5
10
```

In [92]:
```
nums=[12,14,18,21,25]

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

```
25
```

In [94]:
```
nums=[12,14,18,21,25,20]

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

```
25
20
```

In [96]:
```
nums=[12,14,18,21,25,20]

for num in nums:
    if num % 5 == 0:
```

```
        print(num)
        break
```

```
25
```

In [98]:
```
nums=[12,14,18,21,20,25]

for num in nums:
    if num % 5 == 0:
        print(num)
        break  #it will print only 1 number then it break
```

```
20
```

In [100…
```
nums=[7,14,18,21,23,27]  #hear there is no number which is divisible by 5 we got

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

In [102…
```
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 [104…
```
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 [106…
```
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') # hear else we dont write in if block but we can writ
```

```
10
20
Not Found
```

In [108…
```
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') # hear else we dont write in if block but we can writ
```

```
10
```

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

In [110…
```
 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 [112…
```
num =13

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

```
prime number
```

In [116…
```
from array import *

arr=array('i',[])

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

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

```
array('i', [6])
array('i', [6, 3])
array('i', [6, 3, 9])
array('i', [6, 3, 9, 7])
array('i', [6, 3, 9, 7, 8])
```

Way of creating array using numpy

In [118…
```
from numpy import*
arr=array([1,2,3,4,5])
print(arr)
type(arr)
```

```
[1 2 3 4 5]
```

Out[118… numpy.ndarray

In [120…
```
print(arr.dtype)
```

int32

In [122…
```
arr =array([1,2,3,4,5.9])
arr
```

Out[122…
```
array([1. , 2. , 3. , 4. , 5.9])
```

In [124…
```
print(arr.dtype)
```

float64

In [128…
```
arr2 =array([1,2,3,4,5.9],float)
arr2
```

Out[128…
```
array([1. , 2. , 3. , 4. , 5.9])
```

In [130…
```
arr3 =array([1,2,3,4,5.9],int)
arr3
```

Out[130…
```
array([1, 2, 3, 4, 5])
```

In [132…
```
import numpy as np
```

In [134…
```
arr4 = np.linspace(0, 16, 10) # break the code between 10 spaces between 0 to 16
arr4
```

Out[134…
```
array([ 0.        ,  1.77777778,  3.55555556,  5.33333333,  7.11111111,
        8.88888889, 10.66666667, 12.44444444, 14.22222222, 16.        ])
```

In [136…
```
arr5 = np.arange(0,10,2) # arange - as range
arr5
```

Out[136…
```
array([0, 2, 4, 6, 8])
```

In [138…
```
arr6 = np.zeros(5)
arr6
```

Out[138…
```
array([0., 0., 0., 0., 0.])
```

In [140…
```
arr7 = np.ones(5)
arr7
```

Out[140…
```
array([1., 1., 1., 1., 1.])
```

In [ ]: