if

if else

if elif else

nested if

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

```
Data Science
```

In [3]:
```
if False:
    print("SK")

#Why It Doesn't Work
#The condition inside if is False.

#Since if only runs when the condition is True, the print("SK") statement never exe

#The program does nothing because False means "do not run this block of code."
```

In [4]:
```
if False:
    print("SK")
print("AFTAB")
```

```
AFTAB
```

In [5]:
```
if True:
    print("SK")
print("AFTAB")
```

```
SK
AFTAB
```

In [6]:
```
if False:
    print("SK")
else:
    print("AFTAB")
```

```
AFTAB
```

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

In [8]:
```
#To print only even number
```

```
x = 4
r = x% 2

if r==0:
    print("EVEN NUMBER")
```

```
EVEN NUMBER
```

In [9]:
```
x = 5
r = x % 2

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

In [10]:
```
x = 5
r = x % 2

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

```
odd number
```

In [11]:
```
x = 6
r = x % 2

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

```
Even number
odd number
```

In [12]:
```
x = 10
r = x%2

if r == 0:
    print('Even Num')
if r == 1:
    print('Odd Num')
```

```
Even Num
```

In [13]:
```
x = 11
r = x%2

if r == 0:
    print('Even Num')
if r == 1:
    print('Odd Num')
```

```
Odd Num
```

In [14]:
```
x = 17
r = x % 2

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

```
if r!=0:
    print('Odd Num')
```

```
Odd Num
```

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 [16]:
```
x = 5
r = x % 2

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

else:
    print('Odd Num')
```

```
Odd Num
```

In [17]:
```
x = 6
r = x % 2

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

else:
    print('Odd Num')
```

```
Even Num
```

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

In [19]:
```
x = 3
r = x % 2

if r == 0:
    print('Even Num')
    if x>5:
        print('Greater Num')
else:
    print('Odd Num')
```

```
Odd Num
```

In [20]:
```
x = 7
r = x % 2

if r == 0:# if this ststement false then inside it's all the statemnt will be skip.
    print('Even Num')
    if x>5:
        print('Greater Num')
else:
    print('Odd Num')
```

```
Odd Num
```

In [21]:
```
x = 10
r = x % 2

if r == 0:
    print('Even Num')
    if x>5:
        print('Greater Num')
else:
    print('Odd Num')
```

```
Even Num
Greater Num
```

In [22]:
```
x = 4
r = x % 2

if r == 0:
    print('Even Num')
    if x > 5:
        print('Greater Num')
    else:
        print('Not Greater')
else:
    print('Odd Num')
```

```
Even Num
Not Greater
```

In [23]:
```
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
```

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 [25]:
```
x = 2

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

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

Two

In [26]:
```
# elif it wont check till the block once you find the output it wont go to next lin
# you can try with multiple parameter 1, 2 & 3 value in x

x = 4

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

four

In [27]:
```
x = 2

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

Two

In [28]:
```
x = 7

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

In [29]:
```
x = 7

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 [30]:
```
x = 4

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

```
four
```

In [31]:
```
x = 10

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 [32]:
```
#short hand if
a = 30
b = 20
if a > b: print("a is greater than b")
```

```
a is greater than b
```

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 1000 times then what you will you cant copy for 1000 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 [34]:
```
print('data science')
print('data science')
print('data science')
print('data science')
print('data science')
```

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

In [35]:
```
i = 1          # initializing

while i<=5:    # condition
    print('data science')
    i = i + 1  # increment
```

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

In [36]:
```
i = 5 #initializing

while i>=1: #condition
    print("SK")
    i = i - 1  #decrement
```

```
SK
SK
SK
SK
SK
```

In [37]:
```
i = 1

while i <= 5:
    print("SK:",i)
    i = i+1
```

```
SK: 1
SK: 2
SK: 3
SK: 4
SK: 5
```

In [38]:
```
i = 5
while i>=1:
    print("SK",i)
    i = i-1
```

```
SK 5
SK 4
SK 3
SK 2
SK 1
```

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

In [40]:
```
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()

    # the output which we got is very lengty but how to make them one line lets ref
```

```
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 [41]:
```
i = 1

while i <=5:
    print('SK', end = "")# when we mention end then new line will not create
    j = 1
    while j<=4:
        print("AFTAB")
        j=j+1
```

```
    i = i+1
    print()
```

```
SKAFTAB
AFTAB
AFTAB
AFTAB

SKAFTAB
AFTAB
AFTAB
AFTAB

SKAFTAB
AFTAB
AFTAB
AFTAB

SKAFTAB
AFTAB
AFTAB
AFTAB

SKAFTAB
AFTAB
AFTAB
AFTAB
```

In [42]:
```
i = 1

while i <=5:
    print('SK', end = "")# when we mention end then new line will not create
    j = 1
    while j<=4:
        print("AFTAB", end = "")
        j=j+1

    i = i+1
    print()
```

```
SKAFTABAFTABAFTABAFTAB
SKAFTABAFTABAFTABAFTAB
SKAFTABAFTABAFTABAFTAB
SKAFTABAFTABAFTABAFTAB
SKAFTABAFTABAFTABAFTAB
```

In [43]:
```
i = 1

while i <=5:
    print(' SK', end = "")# when we mention end then new line will not create
    j = 1
    while j<=4:
        print(" AFTAB", end = "")
        j=j+1
```

```
    i = i+1
    print()
```

```
SK AFTAB AFTAB AFTAB AFTAB
SK AFTAB AFTAB AFTAB AFTAB
SK AFTAB AFTAB AFTAB AFTAB
SK AFTAB AFTAB AFTAB AFTAB
SK AFTAB AFTAB AFTAB AFTAB
```

In [44]:
```
# lets use while loop usig some numbers

i = 1

while i <= 2 :
    j = 0
    while  j <= 2 :
        print(i*j, end=" ")
        j += 1
    print()
    i += 1
```

```
0 1 2
0 2 4
```

In [45]:
```
# lets use while loop usig some numbers
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 iteration or certaion some condition but for loop it will work with sequence (list, string,int)

In [47]:
```
name = "SK"

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

```
S
K
```

In [48]:
```
#i want print the value individualy
name1 = [1,3,5,'sk']
for i in name1:
    print(i)
```

```
1
3
5
sk
```

In [49]:
```
for i in [2,3,7,8,'hi']:
    print (i)
```

```
2
3
7
8
hi
```

In [69]:
```
range(5)
```

Out[69]:
```
range(0, 5)
```

In [67]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [71]:
```
for i in range(2,5):
    print(i)
```

```
2
3
4
```

In [73]:
```
for i in range(1,20,3):
    print(i)
```

```
1
4
7
10
13
16
19
```

In [77]:
```
for i in range (1,21):
    print(i)
```

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

In [75]:
```
# print the value which is divisible by 5

for i in range(1,51):

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

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

In [79]:
```
# print the value which is not divisible by 5.

for i in range(1,51):

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

1
2
3
4
6
7
8
9
11
12
13
14
16
17
18
19
21
22
23
24
26
27
28
29
31
32
33
34
36
37
38
39
41
42
43
44
46
47
48
49

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 [104…
```
#write the code user ask chocklet from vendor machine ,write the basic code.

x = int(input("How many Chocklets you want?"))

i = 1
while i <=x:
    print('Chocklet')
    i+=1
```

```
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
```

In [100…
```
#write the code user ask chocklet from vendor machine ,write the basic code.

x = int(input("How many Chocklets you want?"))

i = 5
while i <=x:
    print('Chocklet')
    i+=1
```

```
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
```

.If the user says i need 10 choclet but vending machine dont have 10 choclate & machine has only 5 choclate so what you do on those scenario

.We have 3 choice now (eiter stop the transaction by you or you can give only 5 choclate) & may be vendor machine display the result as we are out of the stock

.Now lets try in the code

In [118…
```
ava = 5 # the machine has only 5 choclet

x = int(input('How many choclets you want:?'))

i = 1
while i<=x:
    print('choclet')
    i += 1
# if you check the user wants 10 choclets  but availabe choclet is 5 but we got out
# in this code we just declare but we dint apply any condition to it
```

```
choclet
choclet
choclet
choclet
choclet
choclet
```

In [120…
```
available_chocklet = 5 # the machine has only 5 choclet

x = int(input('How many chocklet you want?'))

i = 1

while i<=x:
    print('Chocklet')
    i+=1
    if i>available_chocklet:
        break
print('Out of stock')
```

```
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Out of stock
```

In [122…
```
available_chocklet = 5 # the machine has only 5 choclet

x = int(input('How many chocklet you want?'))

i = 1
```

```
while i<=x:
    if i>available_chocklet:
        break
    print('Chocklet')
    i+=1
print('Out of stock')
```

```
Chocklet
Chocklet
Chocklet
Chocklet
Chocklet
Out of stock
```

In [124…

```
available_choclet = 5 # the machine has only 10 candis

x = int(input('How many choclets you want:?'))

i = 1
while i<=x:

    if i>available_choclet: # we stop the execution but which code execution not en
        print('out of stock')
        break # break is statement | means jump out of the loop
    print('choclet')
    i += 1

print('bye for now')
```

```
choclet
choclet
choclet
choclet
choclet
out of stock
bye for now
```

In [126…

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

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

i dont want 11 number i want only 5 number for the range of 1 to 10

In [129…

```
for i in range(1,11):
    if i == 6:
```

```
        break
    print(i)
```

```
1
2
3
4
5
```

in continue loop wont be terminate & exclue the assign number it give you entire output

In [132…

```
for i in range(1,11):
    if i == 3:
        continue
    print(i)
```

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

In [138…

```
for i in range(11):
    if i == 6:
        continue
    print('hello:',i)
```

```
hello: 0
hello: 1
hello: 2
hello: 3
hello: 4
hello: 5
hello: 7
hello: 8
hello: 9
hello: 10
```

PASS Statement - pass the code & it wont go (code give you the error)

In [141…

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

```
  Cell In[141], line 1
    for i in range(1,11):
                         ^
SyntaxError: incomplete input
```

In [143…
```
for i in range(1,11):
    pass
```

you need to print the number from 1 to 50 but dont print the number which is divisible by 3 or 5

In [156…
```
for i in range(1,51):

    if i%3 != 0:
        print(i)
print('end')
```

```
1
2
4
5
7
8
10
11
13
14
16
17
19
20
22
23
25
26
28
29
31
32
34
35
37
38
40
41
43
44
46
47
49
50
end
```

In [7]:
```
for i in range(1,20):

    if i%3 != 0 or i%5 == 0:
        print(i)
print('end')
```

```
1
2
4
5
7
8
10
11
13
14
15
16
17
19
end
```

In [9]:
```
for i in range(1,21):

    if i%3 != 0 or i%5 == 0:
        continue
    print(i)
#print('end')
# it will skip all the value which is divisible by 3 or 5
```

```
3
6
9
12
18
```

In [210…
```
for i in range(1,20):
    if i%3 == 0 and i%5 == 0:
        print(i)
print('end')
# when you apply and you wont get the value which is divisible by both 3 & 5 (15)
```

```
15
end
```

In [15]:
```
for i in range(1,20):
    if i%3 == 0 and i%5 != 0:
        continue
    print(i)
print('end')
# when you apply and you wont get the value which is divisible by both 3 & 5 (15)
```

```
1
2
4
5
7
8
10
11
13
14
15
16
17
19
end
```

In [174…

```python
for i in range(1,51):

    if i%3 == 0:
        print(i)
```

```
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
```

In [178…

```python
for i in range(1,20):

    if i%3 == 0:
        continue
    print(i)
print('end')
```

```
1
2
4
5
7
8
10
11
13
14
16
17
19
end
```

In [180…

```
for i in range(1,20):

    if i%3 == 0:
        break
    print(i)
print('end')
```

```
1
2
end
```

In [182…

```
for i in range(1,20):

    if i%3 == 0:
        pass #skiped the condition or block.
    print(i)
print('end')
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
end
```

In [43]:

```
# i dont want to print the values which are even numbers that means print only odd

for i in range (1,21):
```

```
    if (i%2==0):
        continue
    else:
        print(i)
print('End')
```

```
1
3
5
7
9
11
13
15
17
19
End
```

PRINTING PATTERN IN PYTHON

In [46]:
```
print('####')
print('####')
print('####')
print('####')
```

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

In [50]:
```
for i in range(1,5):
    #i=i+1
    print('####')
```

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

What is Iteration? Iteration means repeating a process multiple times. In programming, iteration happens when a loop repeats a block of code until a condition is met.

In [53]:
```
for i in range(1,5):
    if i<=5:
        print('####')
```

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

In [55]:
```
for k in range(4):
    print('#')
```

```
#
#
#
#
```

In [77]:
```
for j in range(4):
    print('# ', end = " ")
```

```
#  #  #  #
```

In [67]:
```
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 [111…
```
for i in range(4):
    for j in range(4):
        print('#', end= " ")
    print()
```

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

In [113…
```
for i in range(4):
    for j in range(4):
        print('#', end= " ")
    print(end=" ")
```

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

In [117…
```
for i in range(1,5):
    for j in range(4):
        if i>j:
            print('#',end=" ")
    print()
```

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

In [119…
```
list(range(5))
```

Out[119…
```
[0, 1, 2, 3, 4]
```

In [131…
```
for i in range(4):
    for j in range(i):
        print('#',end= "  ")
    print()
```

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

In [133…
```
for i in range(4):
    for j in range(i+1):
        print('#', end="  ")
    print()
```

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

In [145…
```
for i in range(4):
    for j in range(4-i):
        print('#', end="  ")
    print()
```

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

In [147…
```
for i in range(1,5):
    print("# "*(5-i))
```

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

for else

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 [150…
```
nums = [12,15,18,21,26]

for num in nums:
```

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

```
15
```

In [154…
```
nums = [12,14,18,21,25,30,35]

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

```
25  30  35
```

In [160…
```
nums = [10,14,18,21,5,10]

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

```
10
```

In [162…
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we got o

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

In [170…
```
nums = [7,14,18,21,23,2]
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
```

In [ ]:
```
nums = [7,14] #hear there is no number which is divisible by 5 we got output as bla
for num in nums:
    if num % 5 == 0:
        print(num)
        break
    else:
        print('Number Not Found') #every iteration it cheking condition
```

In [199…
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we got o

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

```
        break
else:
        print('Number Not Found') # hear else we dont write in if block but we can
```

```
Number Not Found
```

In [192…

```
nums = [10,14,18,21,20,27]

for num in nums:
    if num % 5 == 0:
        print(num)
        break
else:
        print('Not Found')
```

```
10
```

In [190…

```
nums = [10,14,18,21,20,27,30]

for num in nums:
    if num % 5 == 0:
            print(num)
            #break
else:
        print('Not Found')
```

```
10
20
30
Not Found
```

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

In [201…

```
num = 14

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

```
Not prime Number
```

In [213…

```
num = 13

for i in range(3,num):
    if num % i == 0:
        print('Not prime Number')
        break
else:
    print('Prime Number')
```

```
Prime Number
```

In [255…
```
num = 13

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

```
Prime Number
```

In [267…
```
num = 4

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

```
Not prime Number
```

Array in Python

In [282…
```
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)
```

```
array('i', [2, 3, 4, 5, 6])
```

In [294…
```
from array import *
arr = array('i',[])

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

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

```
array('i', [0, 1, 2, 3])
```

In [305…
```
from  array import *
arr = array('i',[])

n = int(input("Enter the lenghth of array:"))

for i in range(n):
    x=int(input("Enter the next value:"))
```

```
    arr.append(x)
print(arr)
type(arr)
```

```
array('i', [2, 3, 4, 5])
```

Out[305… array.array

Way of creating array using numpy

In [311…
```
from numpy import *

arr1 = array([1,2,3,4,5])
print(arr1)
type(arr1)
```

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

Out[311… numpy.ndarray

In [313…
```
print(arr1.dtype)
```

```
int32
```

In [317…
```
arr2 = array([1,2,3,4,6,9],float)
arr2
```

Out[317… array([1., 2., 3., 4., 6., 9.])

In [325…
```
arr3 = array([1,2,3,4,5,6],bool)
arr3
```

Out[325… array([True, True, True, True, True, True])

In [327…
```
import numpy as np
```

In [329…
```
arr4 = np.linspace(0,16,10) # break the code between 10 spaces between 0 to 16 but
                            #because we break into parts thats why it is floats.
arr4
```

Out[329…
```
array([ 0.        ,  1.77777778,  3.55555556,  5.33333333,  7.11111111,
        8.88888889, 10.66666667, 12.44444444, 14.22222222, 16.        ])
```

In [331…
```
arr5 = np.arange(0,10,2) # arange - as range
arr5
```

Out[331… array([0, 2, 4, 6, 8])

In [335…
```
arr6 = np.zeros(5)
arr6
```

Out[335… array([0., 0., 0., 0., 0.])