In [1]:
```
if True: # indentiation is always 4 spaces
    print('Data Science')
```

```
Data Science
```

In [2]:
```
if False:
    print('Data Science')
print('bye for now')
```

```
bye for now
```

In [3]:
```
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 divide by 2 then reminder is 0 then it is even number if reminder is not 0 then it is odd number

In [4]:
```
#to print only even number
x = 4
r = x % 2
if r == 0:
    print('Even number')
```

```
Even number
```

In [5]:
```
#to print only even number
x = 5
r = x % 2
if r == 0:
    print('Even number')
```

In [6]:
```
x = 5
r = x % 2
if r == 0:
    print('Even number')
print('odd number')
```

```
odd number
```

In [7]:
```
x = 8
r = x % 2
if r == 0:
    print('Even number')
print('odd number')
```

```
Even number
odd number
```

In [8]:
```
x = 8
r = x % 2
if r == 0:
    print('Even number')
if r == 1:
    print('odd number')
```

```
Even number
```

In [9]:
```
x = 7
r = x % 2
if r == 0:
    print('Even number')
if r == 1:
    print('odd number')
```

```
odd number
```

In [10]:
```
x = 15
r = x % 2
if r == 0:
    print('Even number')
if r != 0:
    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 [11]:
```
x = 5
r = x % 2
if r == 0:
    print('Even number')
else:
    print('Odd Number')
```

```
Odd Number
```

In [12]:
```
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 [13]:
```
x = 3
r = x % 2
if r == 0:
    print(' Even number')
    if x>5:
        print('greater number')
else:
    print('Odd Number')
```

```
Odd Number
```

In [14]:
```
x = 4
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
```

In [15]:
```
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 [100]:
```
#when you use if it will check all condition but if we mention elif then it wont check all condition
# when we use if condition it will check all every block of code  better debug in pycharm
# you can debug with value 1 & d for both if & elif

x = 3

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

```
Three
```

In [101]:
```
# elif it wont check till the block once you find the output it wont go to next line
# you can try with multiple parameter 1, 2 & 3 value in x

x = 1

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

```
one
```

In [102]:
```
x = 7

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

In [20]:
```
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 [21]:
```
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 [103]:
```
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
```

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 [23]:
```
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 [24]:
```
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 [25]:
```
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 [26]:
```
i = 1         # initializing
while i<=5:    # condition
    print('data science',i)
    i = i + 1  # increment
```

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

In [27]:
```
i = 5          # initializing
while i>=1:    # condition
    print('data science',i)
    i = i - 1  # decrement
```

```
data science 5
data science 4
data science 3
data science 2
data science 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 [28]:
```
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 refer to below code
```

```
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 [29]:
```
i = 1
while i<=5:
    print(' datascience', end = "") # when we mention end then new line will not create
    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]:
```
# 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 [31]:
```
# 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 [32]:
```
name = 'nit'
for i in name:
    print(i)
```

```
n
i
t
```

In [33]:
```
name1 = [1,3.5,'hallo'] #i want print the value individualy
for i in name1:
    print(i)
```

```
1
3.5
hallo
```

In [34]:
```
for i in [2, 3, 7.8, 'hi']:
    print(i)
```

```
2
3
7.8
hi
```

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

```
0
1
2
3
4
```

In [36]:
```
for i in range(1,5):
    print(i)
```

```
1
2
3
4
```

In [37]:
```
for i in range(1,10,3):
    print(i)
```

```
1
4
7
```

In [38]:
```
# print the value which is divisible by 5
for i in range(1,21):
    if i%5==0 :
        print(i)
```

```
5
10
15
20
```

In [39]:
```
# print the value which is divisible by 5 i dont want that value
for i in range(1,21):
    if i%5!=0 :
        print(i)
```

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

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 [40]:
```
# write the code user ask chocklet from vendor machne write the basic code

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

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

```
How many choclets you want:?7
choclet
choclet
choclet
choclet
choclet
choclet
choclet
```

If the user says i need 100 choclet but vending machine dont have 100 choclate & machine has only 50 choclate so what you do on those scenario We have 3 choice now (eiter stop the transaction by you or you can give only 50 choclate) & may be vendor machine display the result as we are out of the stock Now lets try in the code

In [41]:
```
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 output as 10 choclet
# in this code we just declare but we dint apply any condition to it
```

```
How many choclets you want:?5
choclet
choclet
choclet
choclet
choclet
```

In [42]:
```
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 entire code , i want to
        break # break is statement | means jump out of the loop
    print('choclet')
    i += 1

print('bye for now')
```

```
How many choclets you want:?4
choclet
choclet
choclet
choclet
bye for now
```

In [43]:
```
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 entire code , i want to 
        print('out of stock')
        break # break is statement | means jump out of the loop
    print('choclet')
    i += 1

print('bye for now')
```

```
How many choclets you want:?7
choclet
choclet
choclet
choclet
choclet
out of stock
bye for now
```

In [45]:
```
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 [46]:
```
for i in range(1,11):
    if i == 5:
        break
    print(i)
```

```
1
2
3
4
```

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

In [47]:
```
for i in range(1,11):
    if i == 3:
        continue
    print(i)
```

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

In [48]:
```
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 (code give you the error)

In [50]:
```
for i in range(1,11):

```

```
  Cell In[50], line 1
    for i in range(1,11):
                         ^
SyntaxError: incomplete input
```

In [51]:
```
for i in range(1,11):
    pass
```

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

In [52]:
```
for i in range(1,50):
    if i%3 == 0:
        print(i)
print('end')
```

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

In [53]:
```
for i in range(1,50):
    if i%3 == 0:
        continue
    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
end
```

In [54]:
```
for i in range(1,50):
    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
```

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

In [55]:
```
for i in range(1,50):
    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
3
4
5
6
7
8
9
10
11
12
13
14
16
17
18
19
20
21
22
23
24
25
26
27
28
29
31
32
33
34
35
36
37
38
39
40
41
42
43
44
46
47
48
49
end
```

In [56]:
```
# i dont want to print the values which are odd numbers that means print only even numbers
for i in range(1,50):

    if (i%2 != 0):
        pass
    else:
        print(i)
print('bye')
```

```
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
bye
```

In [57]:
```
print('# # # #')
print('# # # #')
print('# # # #')
print('# # # #')
```

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

In [58]:
```
for j in range(4):
    print('#')
```

```
#
#
#
#
```

In [59]:
```
for j in range(4):
    print('#', end=" ")
```

```
# # # #
```

In [60]:
```
for j in range(4):
    print('#', end=" ")

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

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

In [61]:
```
for j in range(4):
    print('#', end=" ")

print()

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

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

In [62]:
```
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 [63]:
```
for i in range(4):
    for j in range(4):
        print('#', end="  ")
    print()
    # pease use debug mode in pycharm
```

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

In [64]:
```
list(range(5))
```

Out[64]:
```
[0, 1, 2, 3, 4]
```

In [65]:
```
for i in range(5):
    for j in range(i):
        print('#', end="  ")
    print()
```

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

In [66]:
```
for i in range(5):
    for j in range(i+1):
        print('#', end="  ")
    print()
```

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

In [67]:
```
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 [68]:
```
nums = [12,15,18,21,26]
for num in nums:
    if num % 5 == 0:
        print(num)
```

```
15
```

In [69]:
```
nums = [12,14,18,21,25,30,35]
for num in nums:
    if num % 5 == 0:
        print(num)
```

```
25
30
35
```

In [70]:
```
nums = [12,14,18,21,25,20]
for num in nums:
    if num % 5 == 0:
        print(num)
```

```
25
20
```

In [71]:
```
nums = [12,14,18,21,25,20]
for num in nums:
    if num % 5 == 0:
        print(num)
        break
```

```
25
```

In [72]:
```
nums = [12,14,18,21,20,25]
for num in nums:
    if num % 5 == 0:
        print(num)
        break
```

```
20
```

In [73]:
```
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 [74]:
```
nums = [10,14,18,21,25,20]
for num in nums:
    if num % 5 == 0:
        print(num)
        continue
```

```
10
25
20
```

In [77]:
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we got output as blank
for num in nums:
    if num % 5 == 0:
        print(num)
        break
```

In [78]:
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we got output as blank
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 [79]:
```
nums = [7,14] #hear there is no number which is divisible by 5 we got output as blank
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
```

In [80]:
```
nums = [7,14,18,21,23,27] #hear there is no number which is divisible by 5 we got output as blank
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 write in for block only
```

```
Number Not Found
```

In [81]:
```
nums = [10,14,18,21,20,27] #hear there is no number which is divisible by 5 we got output as blank
for num in nums:
    if num % 5 == 0:
        print(num)
        #break
else:
        print('Not Found')
```

```
10
20
Not Found
```

In [82]:
```
nums = [10,14,18,21,20,27,30] #hear there is no number which is divisible by 5 we got output as blank
for num in nums:
    if num % 5 == 0:
        print(num)
        #break
else:
        print('Not Found')
```

```
10
20
30
Not Found
```

In [83]:
```
nums = [10,14,18,21,20,27] #hear there is no number which is divisible by 5 we got output as blank
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 [84]:
```
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 [85]:
```
num = 13
for i in range(2,num):
    if num % i == 0:
        print('Not prime Number')
        break
else:
    print('Prime Number')
```

```
Prime Number
```

Array in python

In [86]:
```
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)
```

```
Enter the length of the array2
Enter the next value3
Enter the next value4
Enter the next value5
Enter the next value6
Enter the next value7
array('i', [3, 4, 5, 6, 7])
```

In [93]:
```
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)
```

```
Enter the length of the array9
Enter the next value8
Enter the next value7
Enter the next value6
Enter the next value5
Enter the next value4
array('i', [8, 7, 6, 5, 4])
```

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

In [ ]: