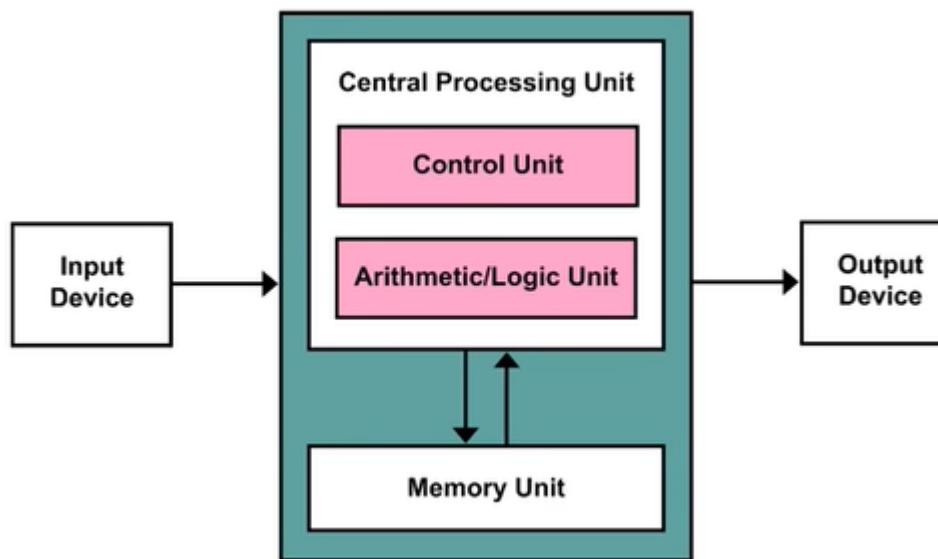


# Sep 3rd,2024



In [ ]: ---CPU has 3 part - CU (Control Unit) | MU(Memory Unit) | ALU (Arithmetic | Logic Unit)

In [ ]: ---- MEMORY UNIT -->  
so far we work in memory unit-that means we save variable in memory & call the variable

In [ ]: - ALU (ARITHMETIC UNIT & LOGIC UNIT) -->  
ARITHMETIC UNIT --> when it comes to Arithmetic unit it performs calculation & we  
LOGICAL UNIT --> when i talk about logical unit which makes your computer to think  
LETS UNDERSTAND THE CONDITION TO COMPUTER TO THINK FOR THAT WE HAVE SOME SPECIAL

if  
if else  
if elif else  
nested if

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

Data Science

In [10]: `if False:`  
`print('Data Science')`  
`print('bye for now')`

bye for now

In [8]: `if True: # indentation is always 4 spaces`  
`print('Data Science')`

```
print('bye for now')
```

Data Science

bye for now

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

In [18]: *#to print only even number*

```
x = int(input('Enter a number'))
r = x % 2

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

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

Odd Number

In [ ]: Above we used two if statements and using multiple if statements can be an issue v

In [22]: *#to print only even number*

```
x = int(input('Enter a Number'))
r = x % 2

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

Even number

In [34]: *#to print only even number*

```
x = 11
r = x % 2

if r == 0:
    print('Even number')
# it prints nothing because it checks and the condition was not satisfied so it st
```

In [1]:

```
x = 5
r = x % 2

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

odd number

In [3]:

```
x = 8
r = x % 2

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

Even number

odd number

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

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

Even number

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

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

odd number

```
In [9]: x = 13
r = x % 2

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

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

odd number

```
In [ ]: if we observe the code its too many line cuz many of the coder always they wanted
        Instead of 2 if we can use if-- else
```

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

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

Even number

```
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 [15]: x = 4
r = x % 2

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

```

    if x>5:
        print('greater number')
    else:
        print('Odd Number')

```

Even number

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

```

In [18]: 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 [20]: 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 [25]: #when you use if it will check all condition but if we mention as elif then it wor
         x = 1

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

```

one

```
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 [29]: x = 5

if(x == 1):
    print('one')
elif(x == 2):
    print('Two')
elif(x == 3):
    print('Three')
elif(x == 4):
    print('four')
# if theres no condition matched we didnt gave any statement to print
```

```
In [31]: 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 [33]: 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 output')
```

wrong output

print('data science')

```
In [37]: print('data science')
print('data science')
```

```
data science
data science
```

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

## While loop & For loop

```
In [43]: 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 [45]: 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 [47]: 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 [49]: 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
```

In [ ]: can we use multiple while loop || nested while loop  
 to understand nested while indepth understand you can use pycharm debug with f8 or

```
In [51]: 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 [53]: i = 1
while i<=5:
    print(' data science', 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()
```

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 [55]: i = 1
while i<=5:
    print(' data science', 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()
```

```
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 [57]: 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
```

In [ ]: FOR LOOP -  
normally while loop it work with condition but for loop it will work with sequence

```
In [59]: name = 'nit'
for i in name:
    print(i)
```

```
n
i
t
```

```
In [61]: name1 = [1,3.5,'hallo']

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

```
1
3.5
hallo
```

```
In [63]: for i in [2, 3, 7.8, 'hi']:
    print(i)
```

```
2
3
7.8
hi
```

```
In [65]: for i in range(5):
    print(i)
```



0  
1  
2  
3  
4

```
In [67]: for i in range(1,5):
         print(i)
```

1  
2  
3  
4

```
In [69]: for i in range(1,10,3):
         print(i)
```

1  
4  
7

```
In [71]: # print the number 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 [73]: # 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

```
In [ ]: --> 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
```

## Break

```
In [75]: for i in range(1,11):  
         print(i)
```

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

```
In [77]: for i in range(1,11):  
         if i == 5:  
             break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT
```

```
In [79]: for i in range(1,11):  
         if i == 5:  
             break #==> WHILE YOU WORK WITH COMPUTER VISION PROJECT  
         print(i)
```

```
1  
2  
3  
4
```

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

## Continue - loop wont be terminate

```
In [84]: for i in range(1,11):  
         if i == 5:  
             continue  
         print(i)
```

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

```
In [86]: 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 [89]: `for i in range(1,11):`

```
Cell In[89], line 1
    for i in range(1,11):
        ^
SyntaxError: incomplete input
```

In [91]: `for i in range(1,11):`  
`pass`

## PRINTING PATTERN IN PYTHON

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

In [95]: `print('# # # #')`  
`print('# # # #')`  
`print('# # # #')`  
`print('# # # #')`

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

In [97]: `for j in range(4):`  
`print('#', end=" ")`

```
# # # #
```

In [99]: `for j in range(4):`  
`print('#', end=" ")`  
  
`for j in range(4):`  
`print('#', end=" ")`

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

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

```
print()

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

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

```
In [103]: 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 [105]: for i in range(4):
           for j in range(4):
               print('#', end=" ")
           print()
           # please use debug mode
```

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

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

```
In [108]: for i in range(5):
           for j in range(i):
               print('#', end=" ")
           print()
```

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

```
In [110]: for i in range(4):
           for j in range(i+1):
               print('#', end=" ")
           print()
```

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

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

```
In [115]: for i in range(4):
           for j in range(4-i):
               print('#', end=" ")
           print()
```

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

```
In [112]: for i in range(4):
           for j in range(4-i):
               print('*', end=" ")
           print()
```

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

## For|Else in python

In [ ]: 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 divisib

```
In [118]: nums = [12,15,18,21,26]

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

15

```
In [120]: nums = [12,14,18,21,25]

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

25

```
In [122]: nums = [12,14,18,21,25,20]
         for num in nums:
             if num % 5 == 0:
                 print(num)
```

25  
20

```
In [124]: nums = [12,14,18,21,25,20]
         for num in nums:
             if num % 5 == 0:
                 print(num)
                 break
```

25

```
In [126]: 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 [128]: 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 [130]: nums = [7,14,18,21,23,27,22] #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('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 [134]: nums = [7,14,18,21,23,27,22] #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('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 [136]: nums = [10,14,18,21,20,27] #hear there is no number which is divisible by 5 we got
for num in nums:
    if num % 5 == 0:
        print(num)
        #break
    else:
        print('Not Found')
```

10  
 20  
 Not Found

```
In [138]: nums = [10,14,18,21,20,27] #hear there is no number which is divisible by 5 we got
for num in nums:
    if num % 5 == 0:
        print(num)
        break
    else:
        print('Not Found')
```

10

```
In [ ]: # prime number - how to check given number is prime number or not
```

```
In [140]: 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 [142]: num = 13

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

Prime Number

```
In [144]: 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', [5])
array('i', [5, 5])
array('i', [5, 5, 1])
array('i', [5, 5, 1, 1])
array('i', [5, 5, 1, 1, 1])
```

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

```
In [148]: from numpy import *
arr = array([1,2,3,4,5])
print(arr)
type(arr)
```

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

```
Out[148]: numpy.ndarray
```

```
In [150]: print(arr.dtype)
```

```
int32
```

```
In [152]: arr = array([1,2,3,4,5.9])
print(arr)
```

```
[1.  2.  3.  4.  5.9]
```

```
In [154]: print(arr.dtype)
```

```
float64
```

```
In [156]: arr2 = array([1,2,3,4,5.9],float)
arr2
```

```
Out[156]: array([1. , 2. , 3. , 4. , 5.9])
```

```
In [158]: arr3 = array([1,2,3,4,5.6],int)
arr3
```

```
Out[158]: array([1, 2, 3, 4, 5])
```

```
In [160]: import numpy as np
```

```
In [162]: arr4 = np.linspace(0, 16, 10) # break the code between 10 spaces between 0 to 16
arr4
```

```
Out[162]: array([ 0.          ,  1.77777778,  3.55555556,  5.33333333,  7.11111111,
  8.88888889, 10.66666667, 12.44444444, 14.22222222, 16.          ])
```

```
In [164]: arr5 = np.arange(0,10,2) # arange - as range
arr5
```

```
Out[164]: array([0, 2, 4, 6, 8])
```

```
In [166]: arr6 = np.zeros(5)
arr6
```



Out[166]: array([0., 0., 0., 0., 0.])

```
In [168]: arr7 = np.ones(5)  
arr7
```

Out[168]: array([1., 1., 1., 1., 1.])

In [ ]: