Class and Object

Class : It's a user defined blueprint or pratatype from which objects are created. It's simply blueprint of object and objects with similarities are considered under one class.

It's defined by using name class keyward:

Ex- class CAR:

Inside class of CAR it's stood by name that we need to put vehicle of personal use ,4 wheels etc inside the class.

we can't put truck, Bike objects in it.

Object : Objects are instances of a class.

Ex- let's consider class CAR of above.

Then it's object can be :

- Gear type.
- No. of Airbags
- Front / back / 4 WD
- Name / Brand

We can access class Attributes using objects of the class

Ex -

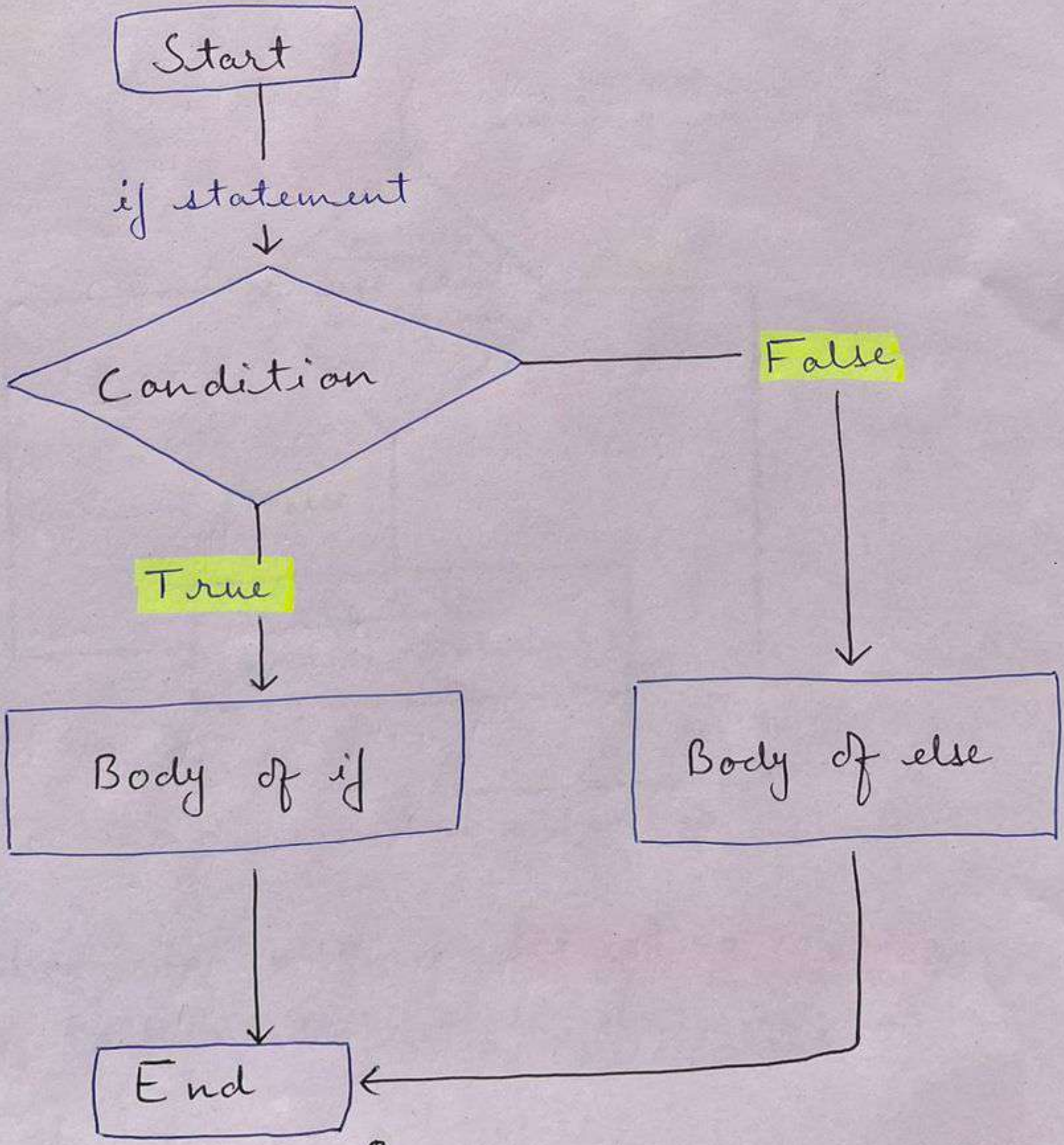
```
Class CAR :            (# creating class of CAR)
      Name   = 'TATA'     [# defining
      Airbag =  6          Attributes]

Car1 = CAR( ) (# creating object of class)

car1.Name    (# accessing Name attribute)
```

* If , Else in python

when we have certain condition to meet then we use if , else , elif in python.



write code to print True if $x = 10$ else False.

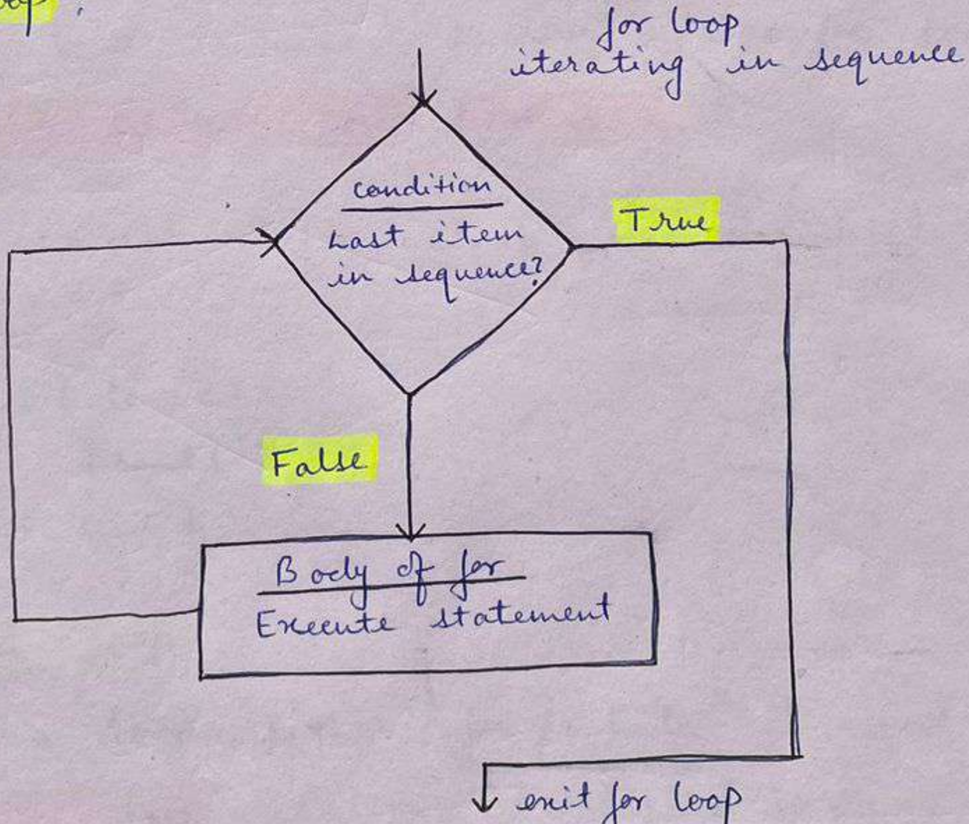
```
x = int(input('Provide natural no'))   [# Input from user]
if x == 10:
    Print ('True')

else:
    Print ('False')
```

* **Loop in python**

Loop is set of sequences of instruction that is continually repeated until a certain condition is reached.

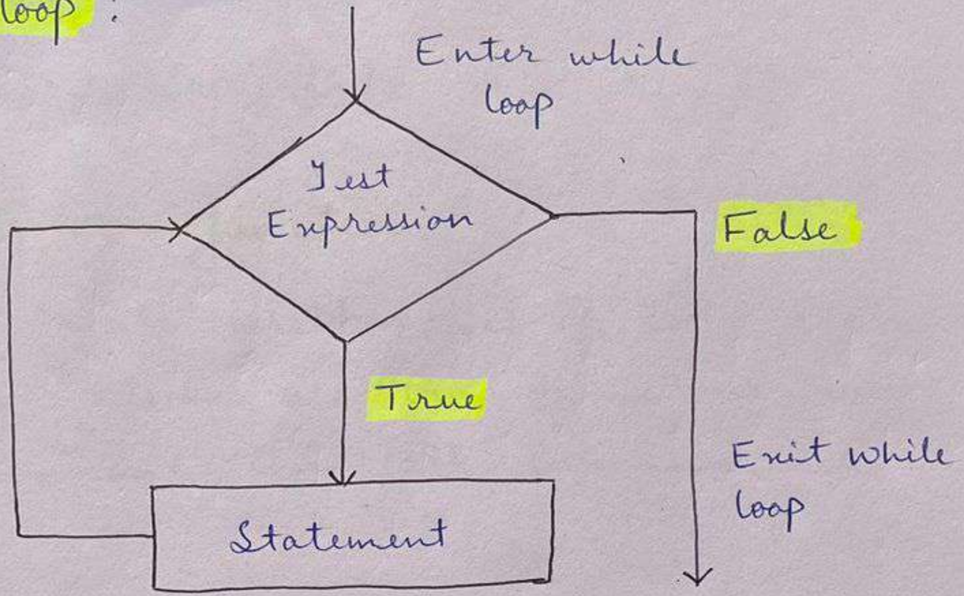
For loop:



A for loop is used for iterating over a sequence. (either a list, tuple, dictonary, set or string)

```
Ex-   Country = ['India', 'USA', 'USSR', 'UK']
      for i in country:
          Print (i)
```

while loop:



while loop is used to run a specific code until a certain condition is meet.

```
Ex-   i = 0          [# initialize the variable]
      n = 5

      while i<=n:
        Print(i)
        i+=1
```

Nested loop:

when a loop either for/while is used inside a loop it's called Nested loop.

Ex- while loop inside for loop, ~~loop~~
for loop inside for loop.

Function in detail:

Function is a block of code which runs when it's called. we can pass data known as parameters into a function.

Function return data as a result.

creating own function:

Rather than writting set of code each time we need, we can define our own function and can call whenever we need.

we can create our own function by using def key-ward.

Ex-
```
def add (a, b, c):
        Print (a + b + c)
```

Above we have created function to add three no. Thus whenever we need to add 3 no.s we need to just call function and pass a, b, c rather than writing every time.

calling our own function:

we just need to call by name and pass arguments

add (2, 3, 4) = 9

Iterator : It's abject used to iterate over an iterable abject using the next() junction.

Iterable : It's abject that one can iterate over by simply passing iter() junction.

- All iterator abjects are iterable but all iterable are nat iterator.

Ex - list, tuple, string is iterable but nat iterator.

let's consider for loop working to better stood.

For loop has three condition which it verifies inside.

- By len ( ) ability it identifies length, thus know where to stap.
- Convert iterable abject to iterator.
- once converted iterate through it and extract the data one by one till length get exhausted.

Generator : It's junction which generate set of data rather than returning it whenever we call it.

We use Generator insteded of print when we have to print huge calculated dataset because print display data once all calculation is over.