In [ ]:
```
# What is a linkedlist.
Linkedlist is a linear datastructure.
It is a collections of nodes.
Nodes:Combination of Data and next node address
    Nodes --> Data|Adress of next Node
For each and every node we will store two things data and address of next node
Head --> First node of a linkedlist is known as head
Tail -->Last node of a linkedlist is known as tail(Address of Tail is always be None)
```

In [ ]:
```
Steps for implementation of linkedlist
1.Create a node --> Data and next
2.Make Connections
```

In [6]:
```
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None
c1=Node(10)
print(c1.data)
print(c1.next)
c2=Node(20)
print(c2.next)
print(c2.data)
c1.next=c2
print(c1.next)
print(c2)
c3=Node(30)
print(c3.data,c3.next)
print(c2.next)
c2.next=c3
print(c2.next,c3)
```

```
10
None
None
20
<__main__.Node object at 0x000001EFB17A9C10>
<__main__.Node object at 0x000001EFB17A9C10>
30 None
None
<__main__.Node object at 0x000001EFB292A730> <__main__.Node object at 0x000001EFB292A730>
```

In [9]:
```
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None
def printll(head):
    while head is not None:
        print(str(head.data)+"-->",end=" ")
        head=head.next
    print("None")

c1=Node(10)
c2=Node(20)
c3=Node(30)
c4=Node(40)
c1.next=c2
c2.next=c3
c3.next=c4
printll(c2)
```

```
20--> 30--> 40--> None
```

In [10]:
```
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None
def printll(head):
    while head is not None:
        print(head.data)
        head=head.next

c1=Node(10)
c2=Node(20)
c3=Node(30)
c4=Node(40)
c1.next=c2
c2.next=c3
c3.next=c4
printll(c2)
```

```
20
30
40
```

In [ ]:
```
#Insertion of node at begining
i will implement a single function that will work for inswertion at begining , end , pos
we can implement sepreate function , begining, pos,end
```

In [ ]:
```
#Insertion of node at begining.
1.Create a node
2.Make the connection with exsiting linkedlist
3.Change the head to the new node

#Insertion at given pos
1.Create a node
2.Take two pointers prev and next node
3.Prev.next=newNode
4.newNode.next=next
5.return head

#Insertion at the last position
1.Create a node
2.use while loop to go till tail node
3.tail.next=newNode
4.return head
```

In [11]:
```
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None
def printll(head):
    while head is not None:
        print(str(head.data)+"-->",end=" ")
        head=head.next
    print("None")
def insertion_at_begin(data,head):
    newNode=Node(data)
    newNode.next=head
    head=newNode
    return head


c1=Node(10)
c2=Node(20)
c3=Node(30)
c4=Node(40)
c1.next=c2
c2.next=c3
c3.next=c4
printll(c1)
x=insertion_at_begin(60,c1)
printll(x)
```

```
10--> 20--> 30--> 40--> None
60--> 10--> 20--> 30--> 40--> None
```

In [15]:
```
#Searching->[10,20,30,40,50]  ele=50   return -1
#Linear Searching
#Given an array arr[] of N elements,
#the task is to write a function to search a given element x in arr[].
x=[10,20,30,40,50,60]
key=40
for i in range(len(x)):
    if x[i]==key:
        print("Element found at",i)
        break


else:
    print("Element not present")
```

```
Element found at 3
Element not present
```

In [ ]:
```
for with break and else --> In that if break statement is not executed then always else part will be executed
```