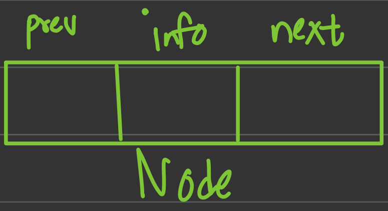
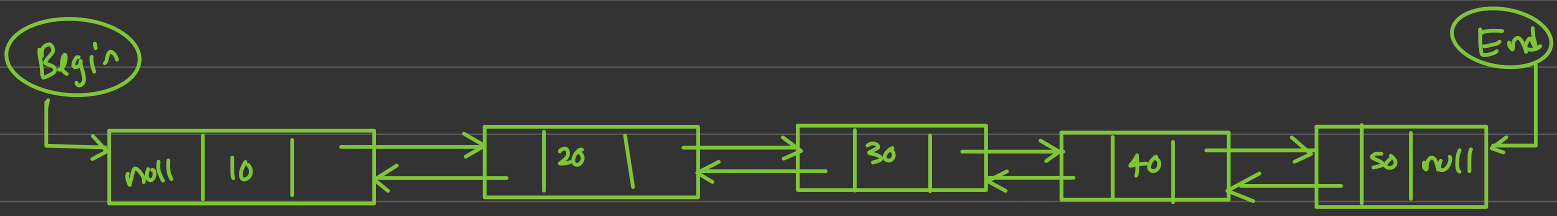
13 June 2023

Doubly LinkedList:





Doubly LL Traversal:

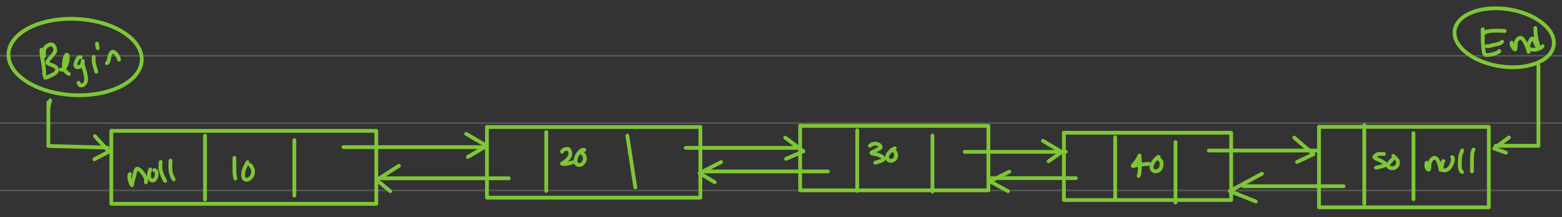
Fwd (Left to Right): same code as Singly LL

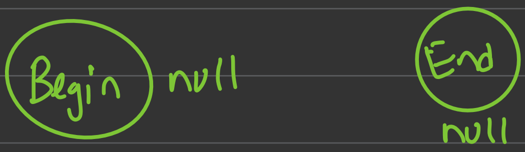
Bwd (Right to Left):

```
1. If Begin = Null
       LL empty
2  Set P = End
   while (p != null)
       { print(p.info)
         p = p.prev
       }
3. End.
```

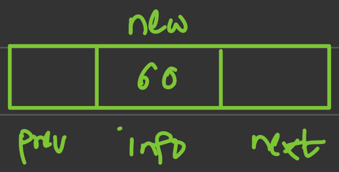
Searching in DLL: exactly same as SLL.

Inserting a node at right end:





1. Create node new

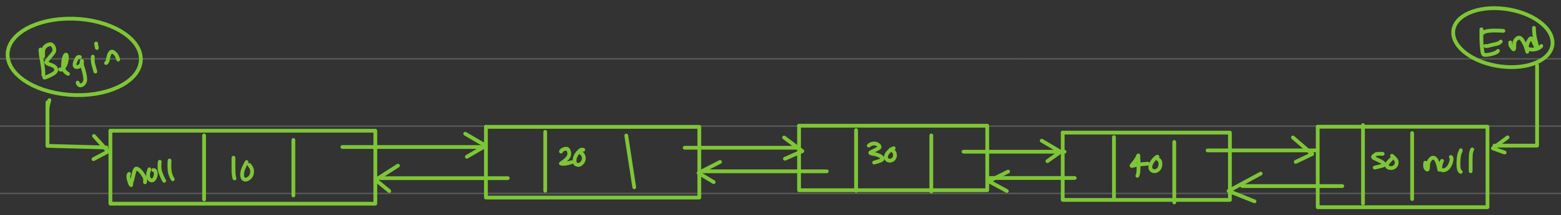


2 new.info = data; // data = 60

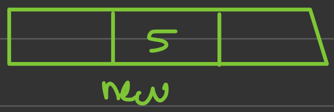
```
3. If Begin = null
       Begin = new
       End = new
       new.next = null
       new.prev = null

4  Set p = End
5  new.next = null
   new.prev = p
   p.next = new
   End = new
```

Insert at Left End.



```
1. Create new
2. new.info = data
```

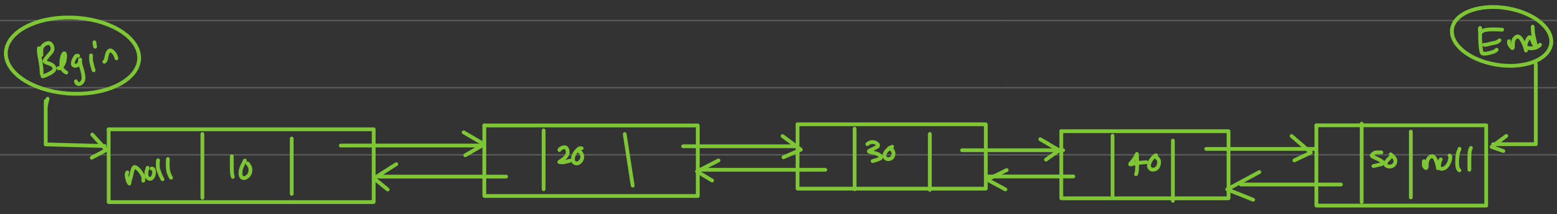


```
3. If Begin = null
        new.next = null
        new.prev = null
        Begin = new
        End = new

4  Set p = Begin

5. new.next = p ; // new.next = Begin
   new.prev = null
   p.prev = new
   Begin = new
```

Delete:



1. Enter data = 30 to be searched

```
2. If Begin = null,
       Sop( " LL empty cannot delete");
   else

3 If Begin.info = = data & Begin.next
     = = null & End.prev = = null
     { Begin = null          → If LL consists of
       End = null              only one node
     } else

4. If Begin.info == data
   node to be deleted is the 1st node
       Set p = Begin
       Begin = Begin.next
       Begin.prev = null
       p = null

5. If End.info == data
   node to be deleted is the last node
       Set p = End
       End = End.prev
       End.next = null
       p = null

6. node to be deleted is middle node
   Set p = Begin.next

   while (p != End)
   { if (p.info == data)
           break
       else
           p = p.next
   }
   → p.prev.next = p.next
     p.next.prev = p.prev
     p = null

7.
   if (p == End)
       Sop(" Node to be deleted not found.")
```