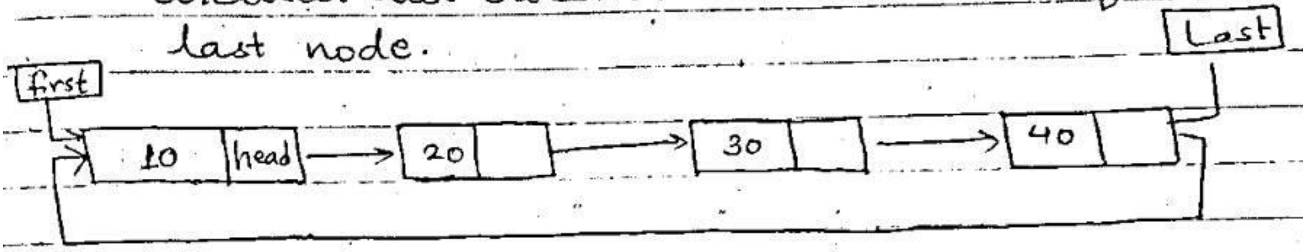
Circular link list

item field -> value
link field -> address

Circular linked list:

A circular linked list is a list where a linked list of last nodes point to the very first node of the list. In circular linked list, the first and final node are linked together. This type of linked list is most useful for managing buffers for data ingest, and in cases were we have one of ~~the~~ object in a list and we ~~should~~ wish to interate through all other objects in the list in one particular order. As because circular list doesnot have a natural first or last node.



A circular link list has no end. However, we set the external pointer ~~head~~ first and last as a linear link list. A head pointer is used for both first and last.

Algorithm to insert a new node at the begining of ~~node~~ linked list:

Step: 1 Create a new node
Step: 2 Assign an item to the new node.
step: 3 If the list is priority empty, set the linked field of the new node to point itself.
Step: 4 Otherwise, set the link field of last node to point new node:

Step : 5 Set the link field of new node to point the first node.

Step : 6 Set the start to point the new node.

Algorithm to insert a new node at the end of the linked list:

Step : 1 Create a new node.

Step : 2 Assign an item to the new node.

Step : 3 If the list is priorily empty, set the linked ~~list~~ field of new node to point itself.

Step : 4 Otherwise, set the linked ~~list~~ field of last node to point the new node.

Step : 5 Set the linked field of new node to point the first node.

Step : 6 Set the external pointer last to point the new node.

Deleting node from the begining:

Algorithm to delete a node from the begining (deleting first node)

~~Step : 1~~

Step : 1 If the list is empty print "empty list" and exit.

Step : 2 Set link field of last node to point the second node (next node of first node)

Step:3 Set start ~~equals~~ to point the second node.
Step:4 End

Algorithm to delete a node from the end (deleting last node)

Step:1 If the linked list is empty print "empty list" and exit.
Step:2: Set the linked field of last node as NULL.
Step:3: Search for the node whose next node contain the NULL value in linked field.
Step:4 When the node is found set the link field of the node to point the first node.
Step:5 Set the external pointer last to point the node.
Step:6 End

Doubly linked list:

It is a list in which all nodes are linked together by multiple number of link which help in accessing both the successor node and predi~~cessot~~cessor node from the given node position. It is bi-directional traversing node.

Each node in doubly linked list has two linked fields. These are used to point the successor and predicessor node.