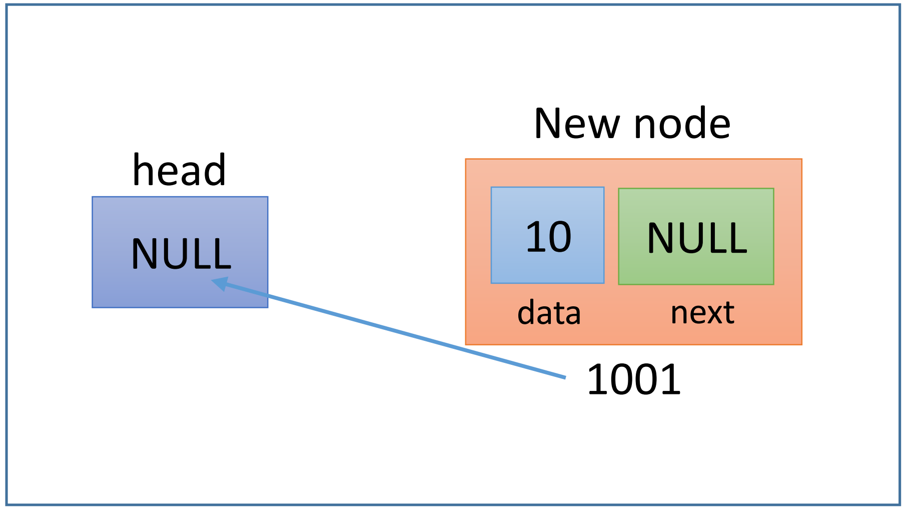
Singly Linked List

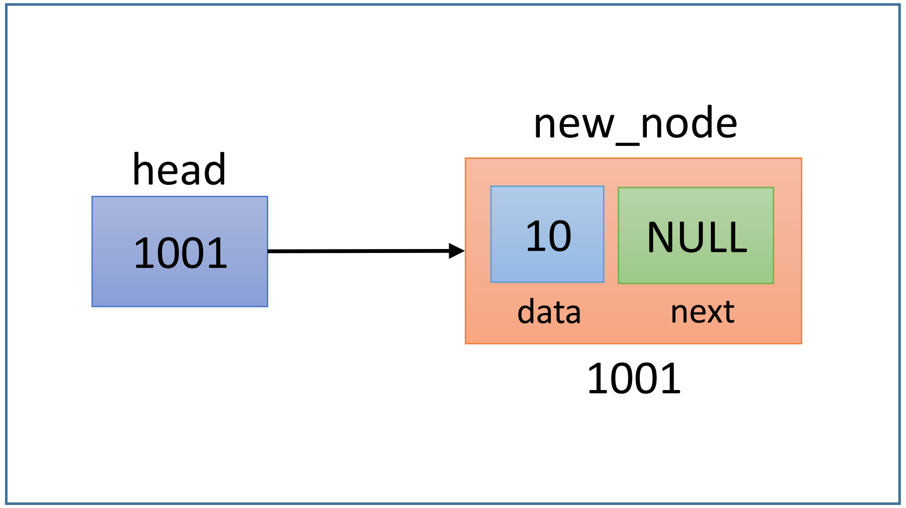
Insertion at first

- **When list is empty**

1. Create a new node.

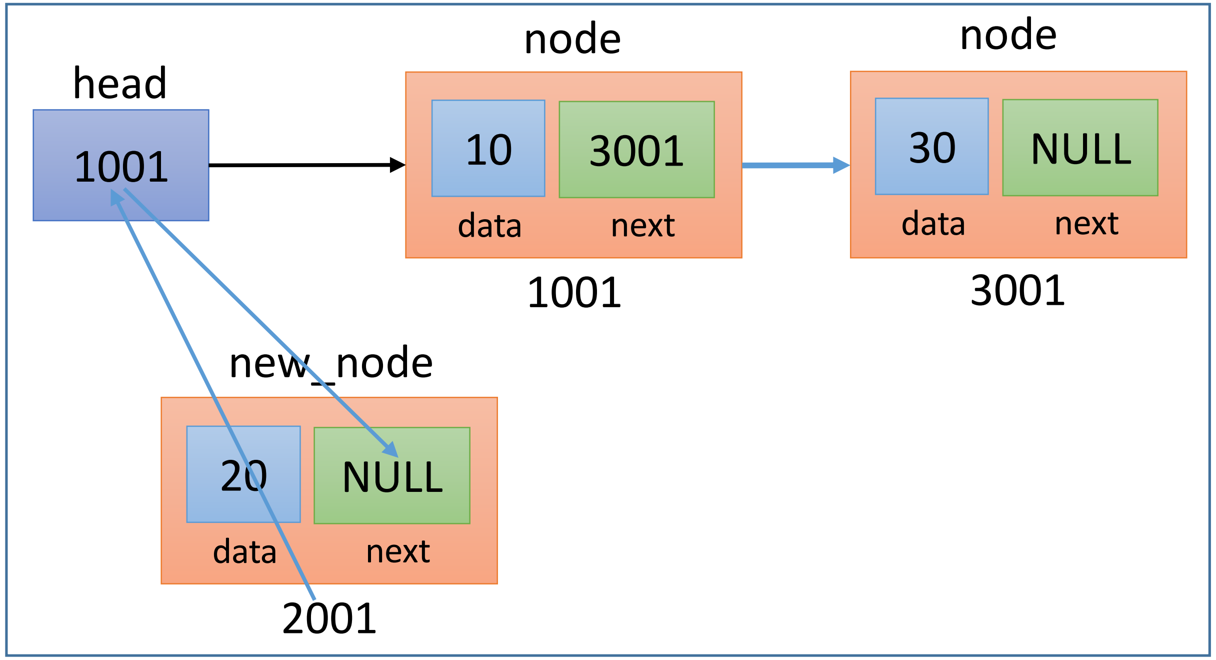
2. Add the new node address in head

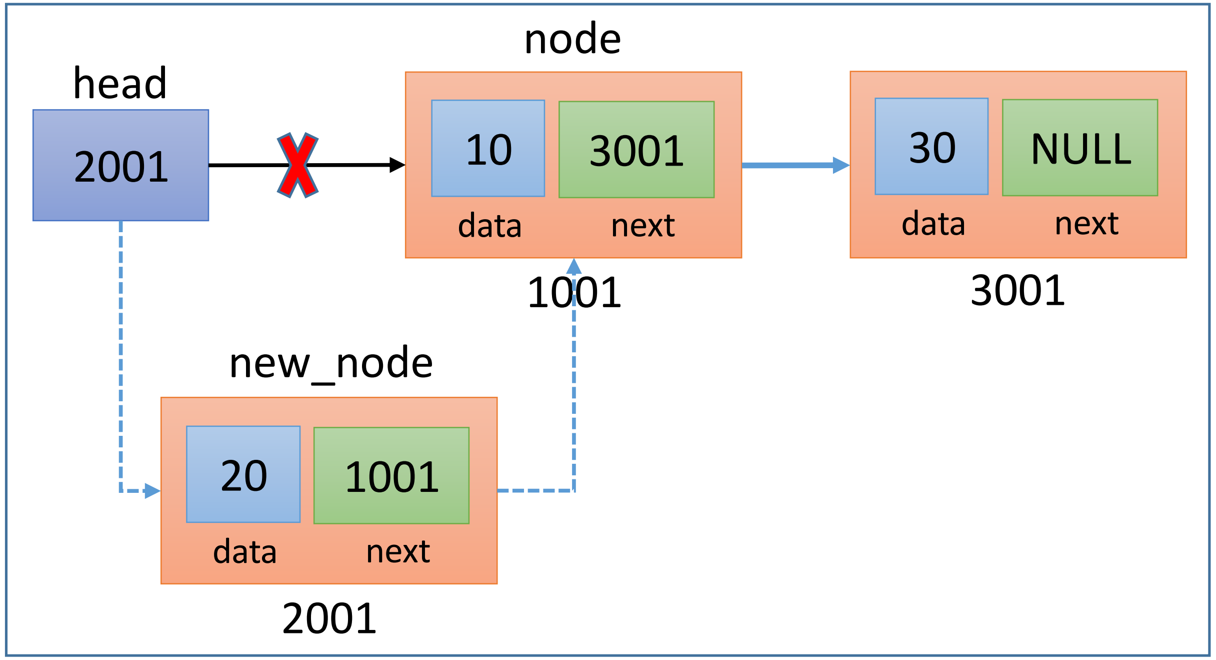




- **When list is not empty**

1. Create a new node.
2. Add head current address in new node next.
3. Add the new node address in head

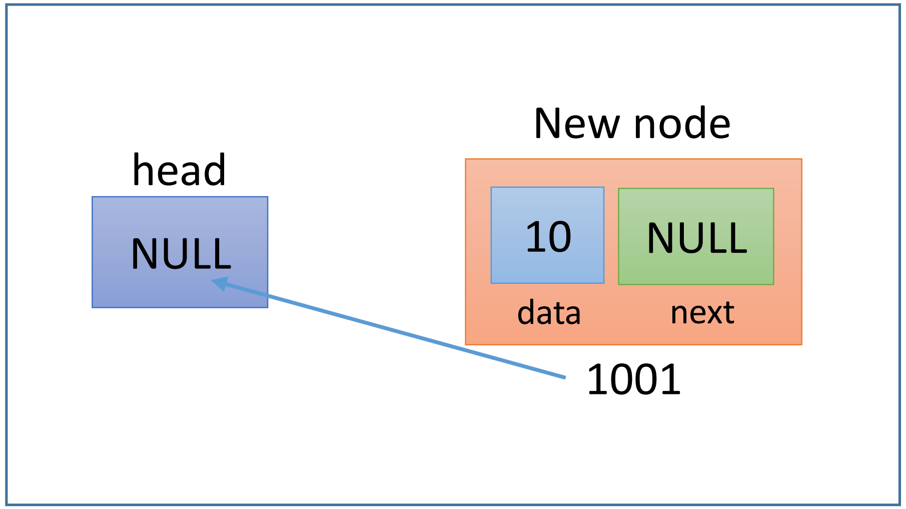


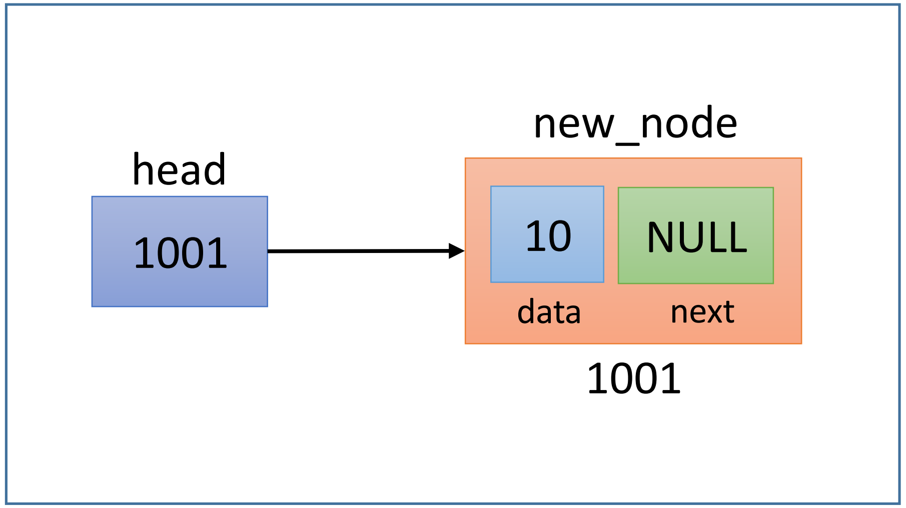


Insertion at last

- **When list is empty**

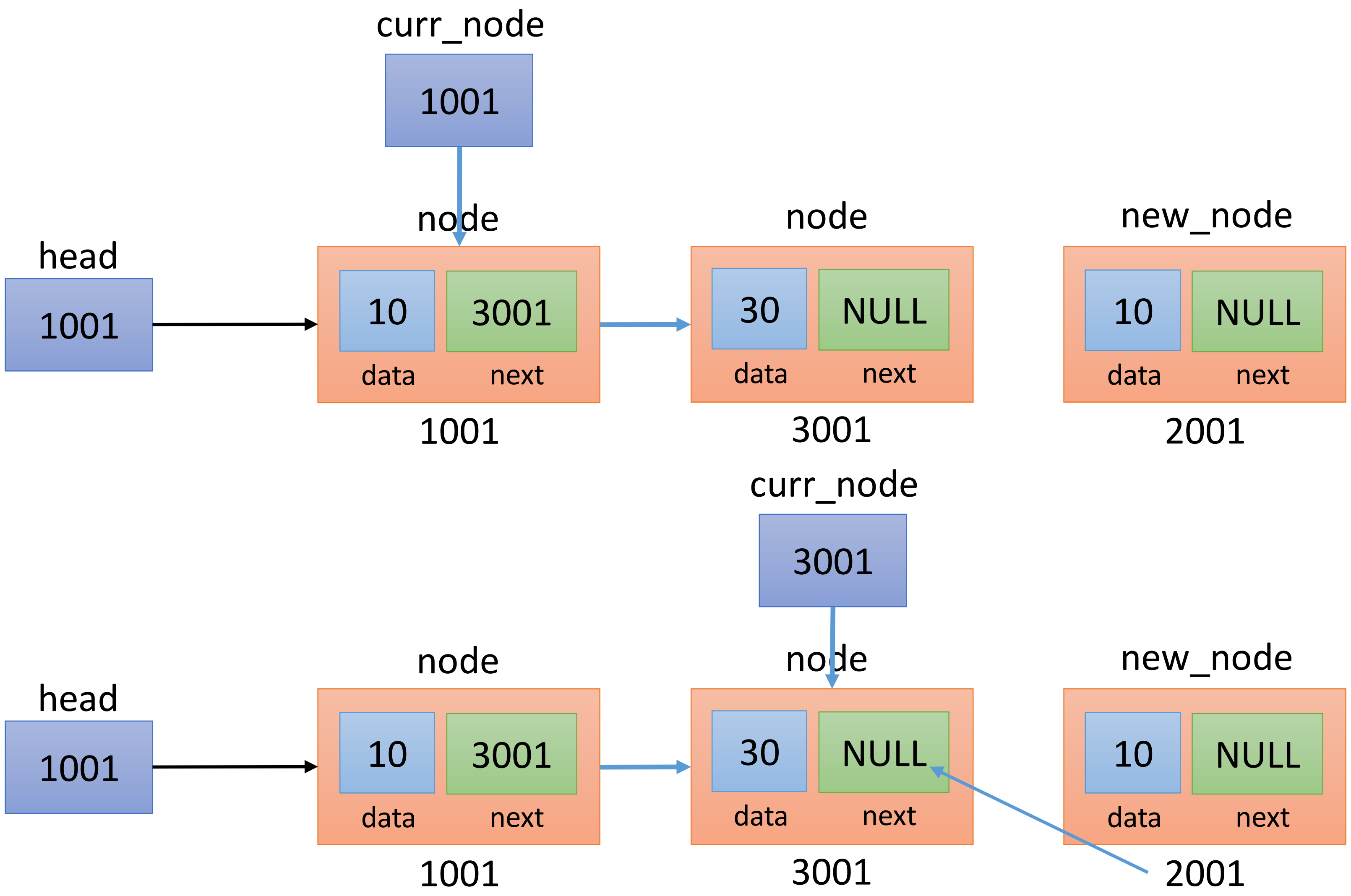
1. Create a new node.
2. Add the new node address in head

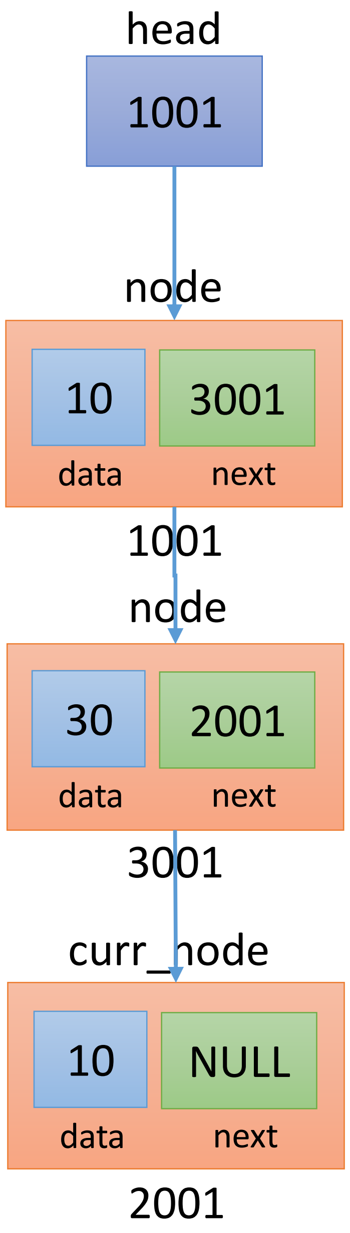




➢ When list is not empty

1. Create a new node.
2. Traverse till the end.
3. Add the new node address in last node next.





Insert after any node

- **When list is empty**

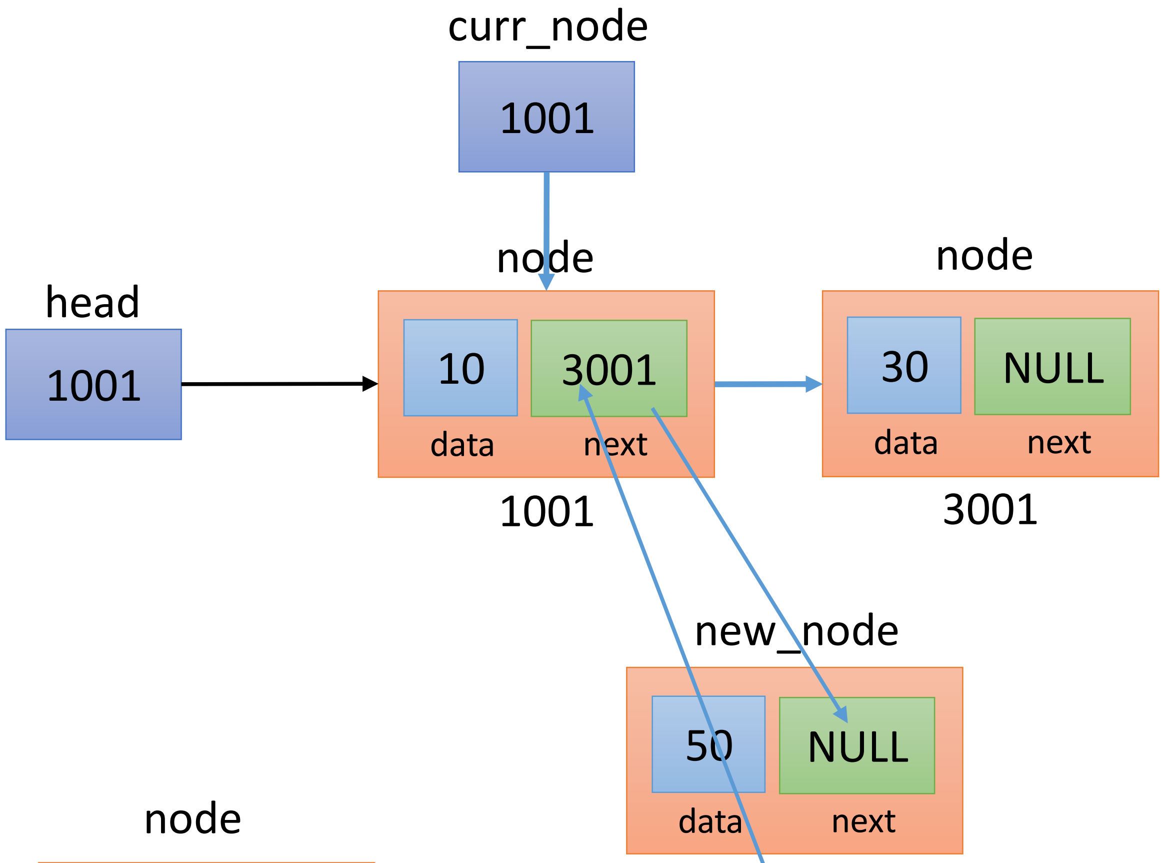
    1. Show the message that List is empty.

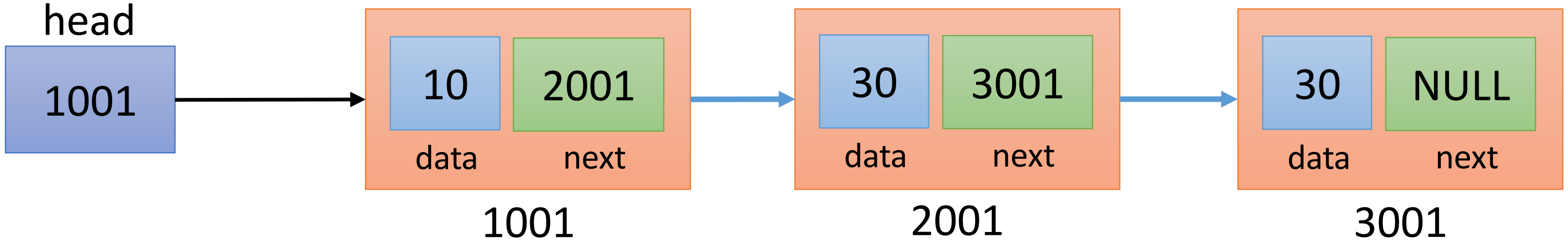
- **When list is not empty**

    - **If node found**

        1. Traverse till the node found

        2. Add current node next in new node next and add new node address in current node next.





    - **If node not found**

        1. Show the message that Node not found.

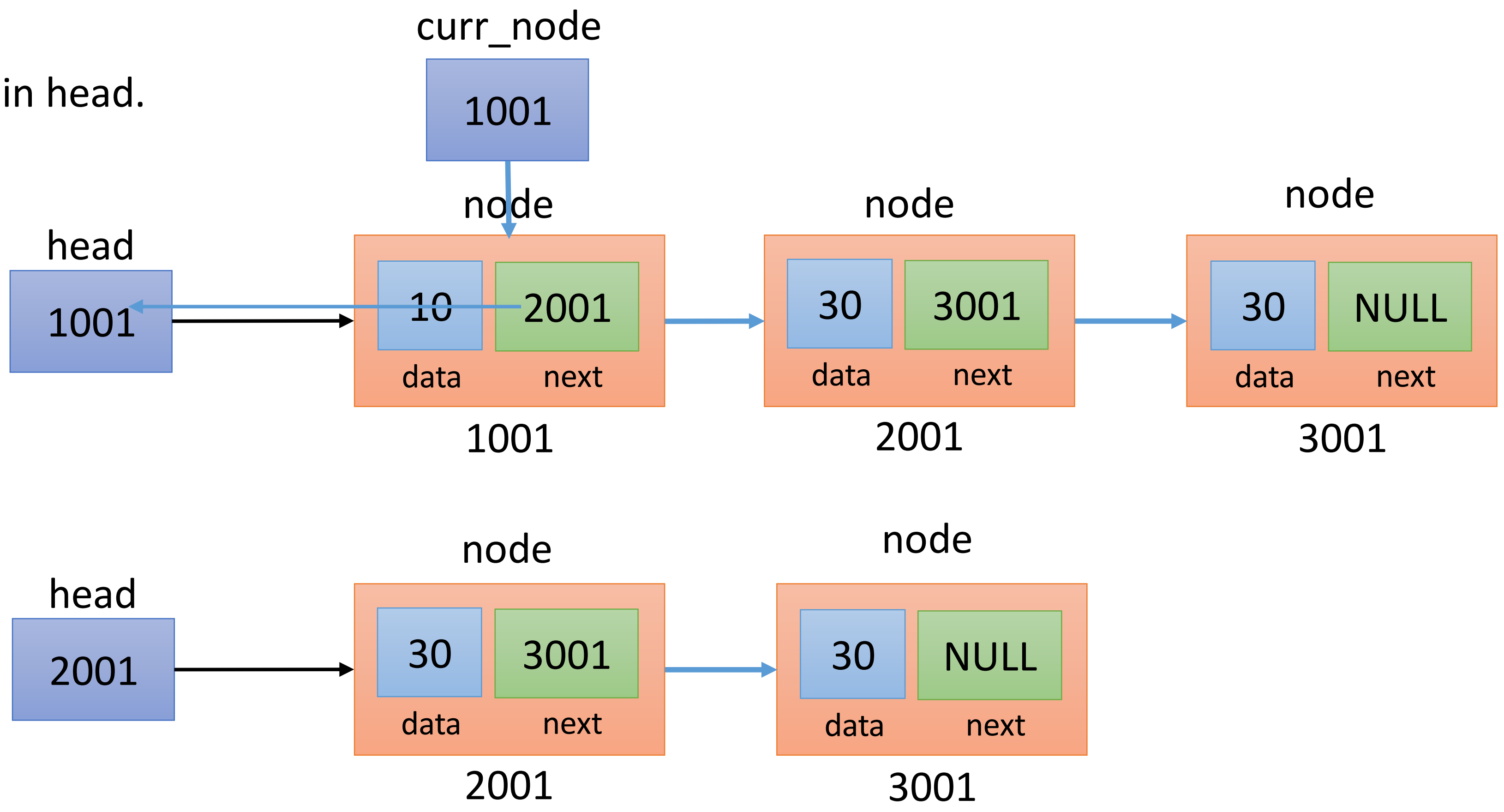
Delete first

- **When list is empty**

  1. Show the message that List is empty.

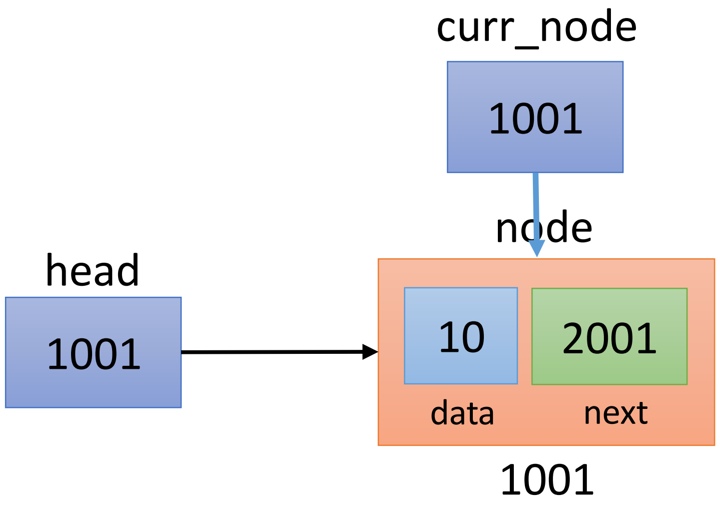
- **When list is not empty**

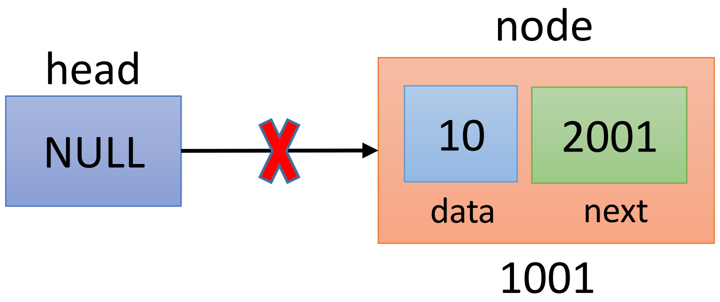
  1. Add current node next in head.
  2. Delete current node.



Delete last

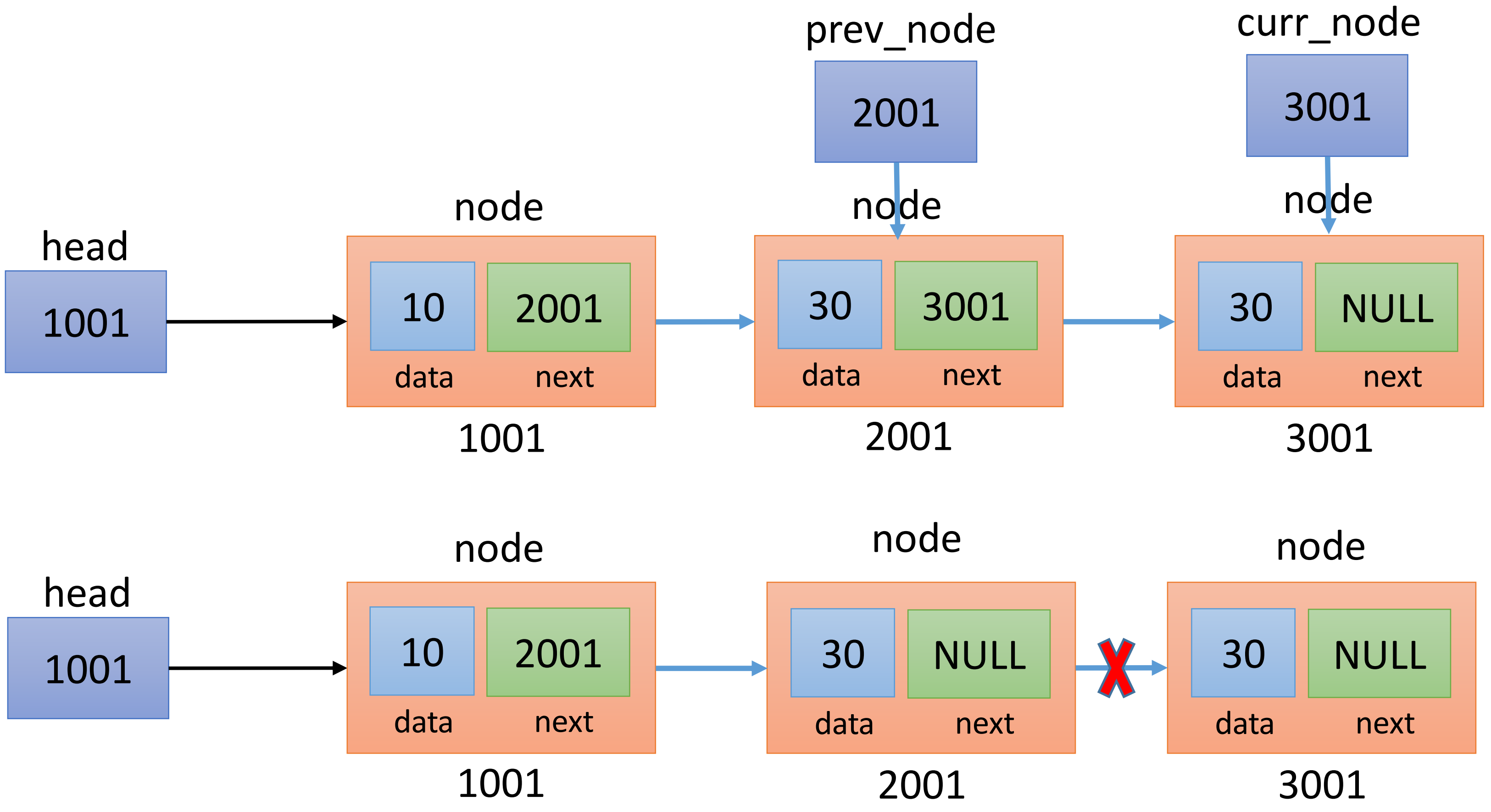
- **When list is empty**
  1. Show the message that List is empty.
- **When list is not empty and have one node**
  1. Set head value to NULL.
  2. Delete current node.





- **When list is not empty and have more than one node**

1. Traverse to the last node.
2. Store the last node previous node in prev_node.
3. Set prev_node next NULL.
4. Delete the last node.



Delete a specific node

- **When list is empty**

  1. Show the message that List is empty.

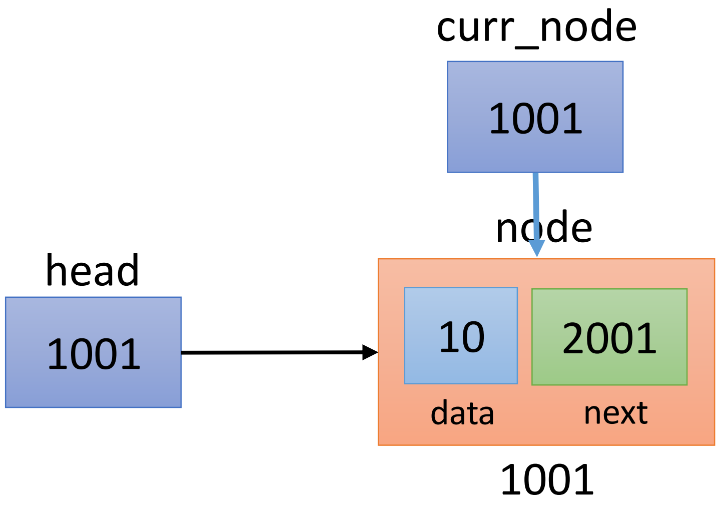
- **When list is not empty and have one node**

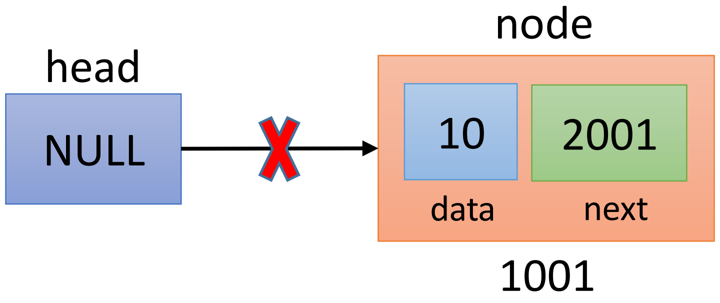
  - **If node found**

    1. Set head value to NULL.
    2. Delete current node.

  - **If node not found**

    1. Show the message that Node not found.



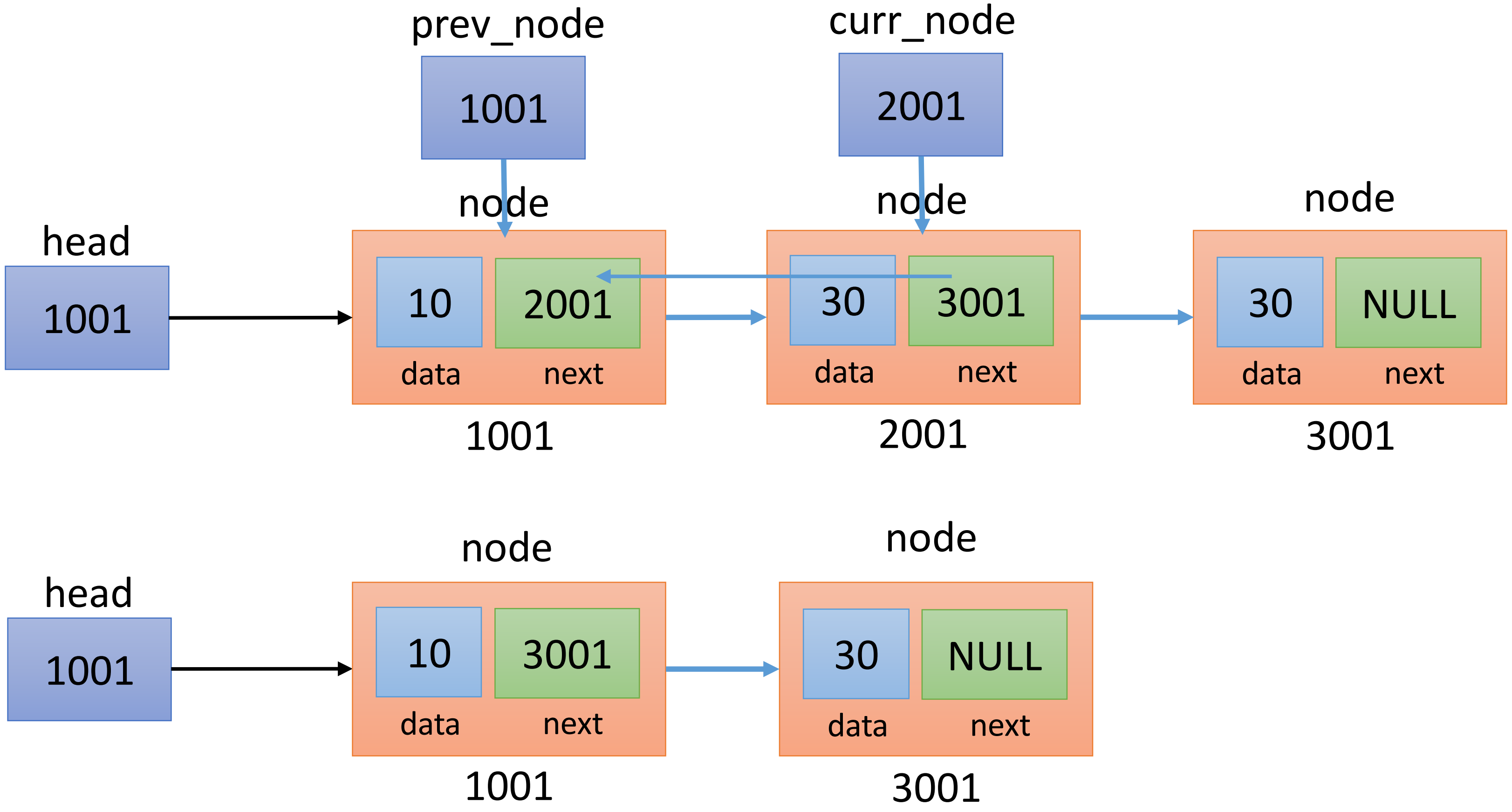


- **When list is not empty and have more than one node**
  - **If node found**

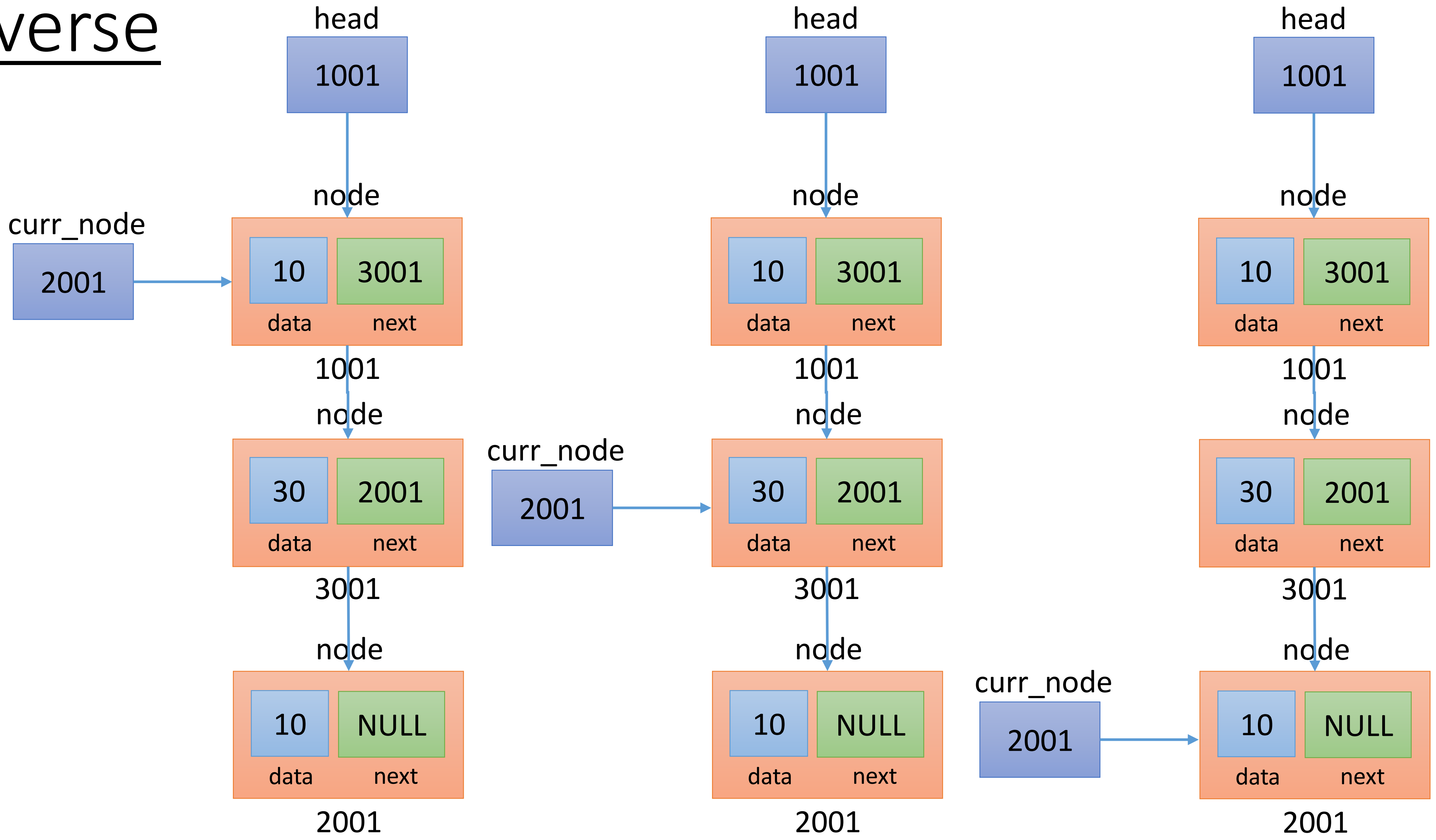
1. Traverse to the required node.
2. Store the required node previous node in prev_node.
3. Set prev_node next NULL.
4. Delete the required node.

- **If node not found**

1. Show the message that Node not found.



Traverse


head
1001
node
curr_node
2001
10
3001
data
next
1001
node
30
2001
data
next
3001
node
10
NULL
data
next
2001
head
1001
node
10
3001
data
next
1001
node
curr_node
2001
30
2001
data
next
3001
node
10
NULL
data
next
2001
head
1001
node
10
3001
data
next
1001
node
30
2001
data
next
3001
node
curr_node
2001
10
NULL
data
next
2001

Update

- **When list is empty**

  1. Show the message that List is empty.

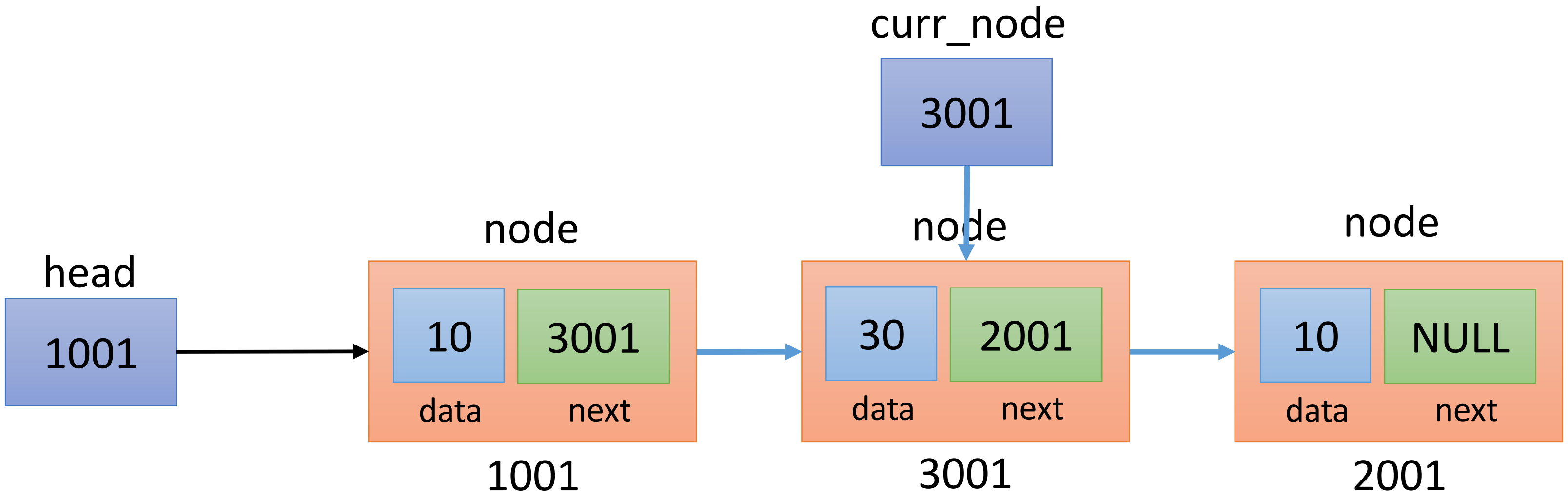
- **When list is not empty**

  - **If node found**

    1. Set current node data with new data.

  - **If node not found**

    1. Show the message that Node not found.


curr_node
3001
head
1001
node
10
3001
data
next
1001
node
30
2001
data
next
3001
node
10
NULL
data
next
2001