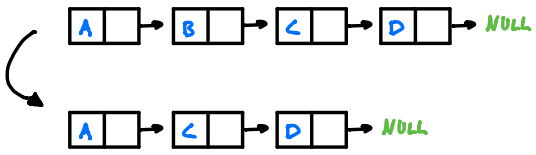
Singular Linked Lists Deletion

Given a key (data field) delete node with this field

Assume elements in the linked list are unique

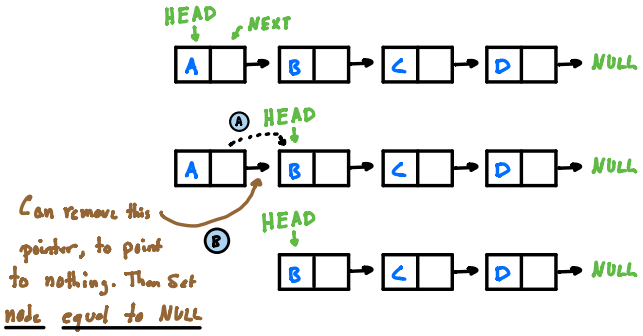
Example: Delete node with data field “B”



2 Cases:

- Node to be deleted is head
- Node to be deleted is not head

Node to be deleted is head



What we have to do:

A move head pointer over to next element in the linked list

B Set former "head" pointing to NULL


```
def delete_node(self, key):
    current_node = self.head
    if current_node and current_node.data == key:
        self.head = current_node.next
        current_node = None
        return
```

Going to start at beginning of the linked list

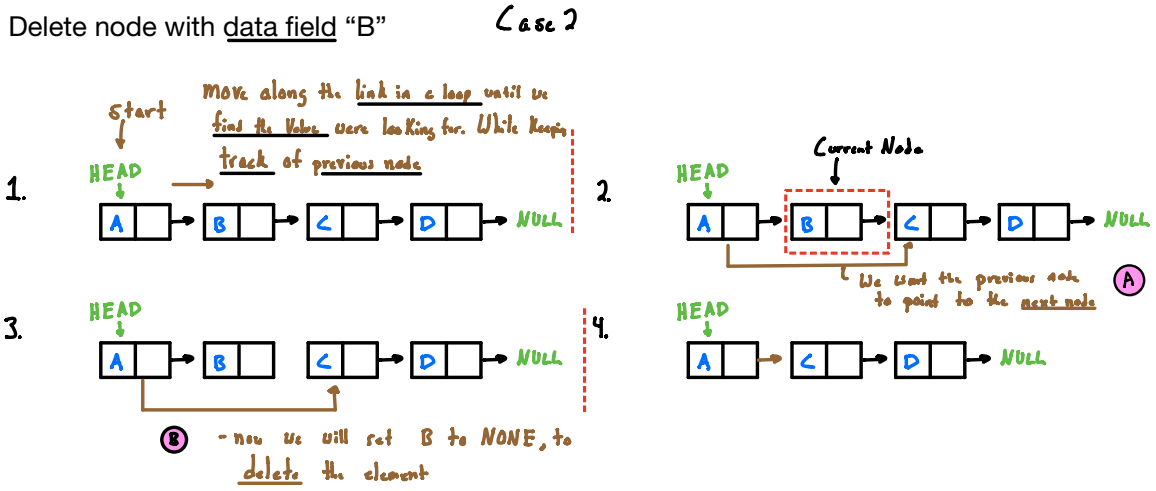
Check if list is not empty and if our current node data is equal to the node to be deleted
(Both are True, so we proceed)

We set the head value equal to the next element in our linked list (A)

Set former "head" pointing to NULL (or NONE) (B)

Node to be deleted is not head

Delete node with data field "B" Case 2




```
def delete_node(self, key):
    current_node = self.head
    if current_node and current_node.data == key:
        self.head = current_node.next
        current_node = None
        return
    previous = None
    while current_node and current_node.data != key:
        previous = current_node
        current_node = current_node.next

    if current_node is None:
        return
    previous.next = current_node.next
    current_node = None
```

Case 1 ABOVE

Going to start at beginning of the linked list

We're keeping track of the Previous node, set initial equal to NONE

Iterate through the linked list until the data field matches our input.

While we iterate through the linked list we will keep track of our previous node and the current node

- Now we've exited the loop

Check if element is in the list

A Set the previous node pointing to the next node

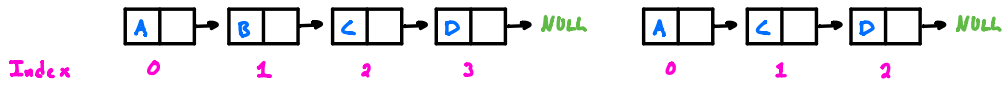
B Set current node (the one we want to delete) to NONE

Delete Node at Position

Given a position, delete node with this position

Assume elements in linked list are unique.

Example: Delete node with position 1




```
def delete_node_at_pos(self, position):
    current_node = self.head
    if position == 0:
        self.head = current_node.next
        current_node = None
        return

    previous = None
    count = 0
    while current_node and count != position:
        previous = current_node
        current_node = current_node.next
        count += 1
    if current_node is None:
        return
    previous.next = current_node.next
    current_node = None
```

Going to start at beginning of the linked list

If were given Position 0, we know were deleting the head node, so we follow the steps above in case 1

Were keeping track of the Previous node, set initial equal to NONE and the Count

While node isnt equal to NONE and not in desired Position were going to loop through the list
- Keeping track of Previous node and current node

Increment the count each loop to track the index
Exit loop when our Position is found

Check if element is in the list

Set the previous node pointing to the next node

Set current node (the one we want to delete) to NONE