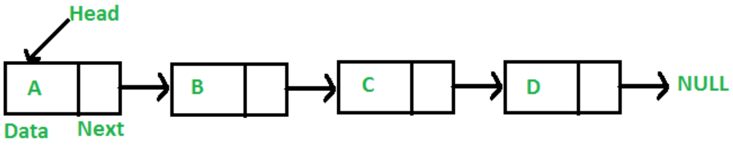
Exercises for Lab 2

Linked Lists



The image above illustrates a LinkedList. Each box represents a Node in this LinkedList. Each node has a ‘data’ field and a ‘next' field. The ‘next’ field holds a reference to the next Node in the chain.

a) Create a **Node** class. The Node should be constructed with one parameter, 'data'. Under construction, the class’s ‘data' field should be set to the parameter ‘data' and the class’s ‘next' field set to `None`. Remember that all classes must be written in CamelCase.

b) The nodes can be linked together to create a chain. Create at least 3 nodes and manually connect them using the “next” attribute in the node object.

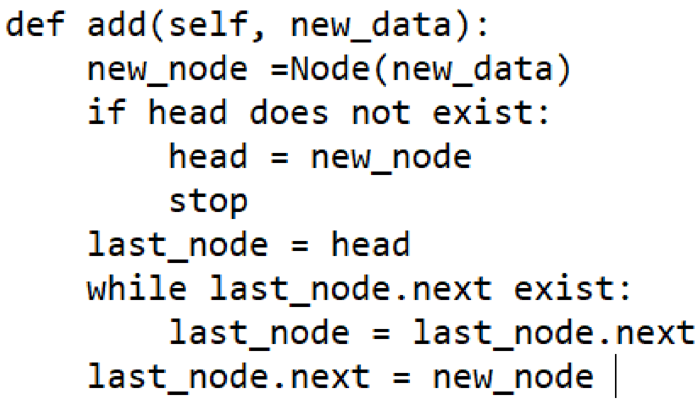
Example:

```
node1 = Node(1)
node2 = Node(2)
node3 = Node(3)

node1.next = ...
```

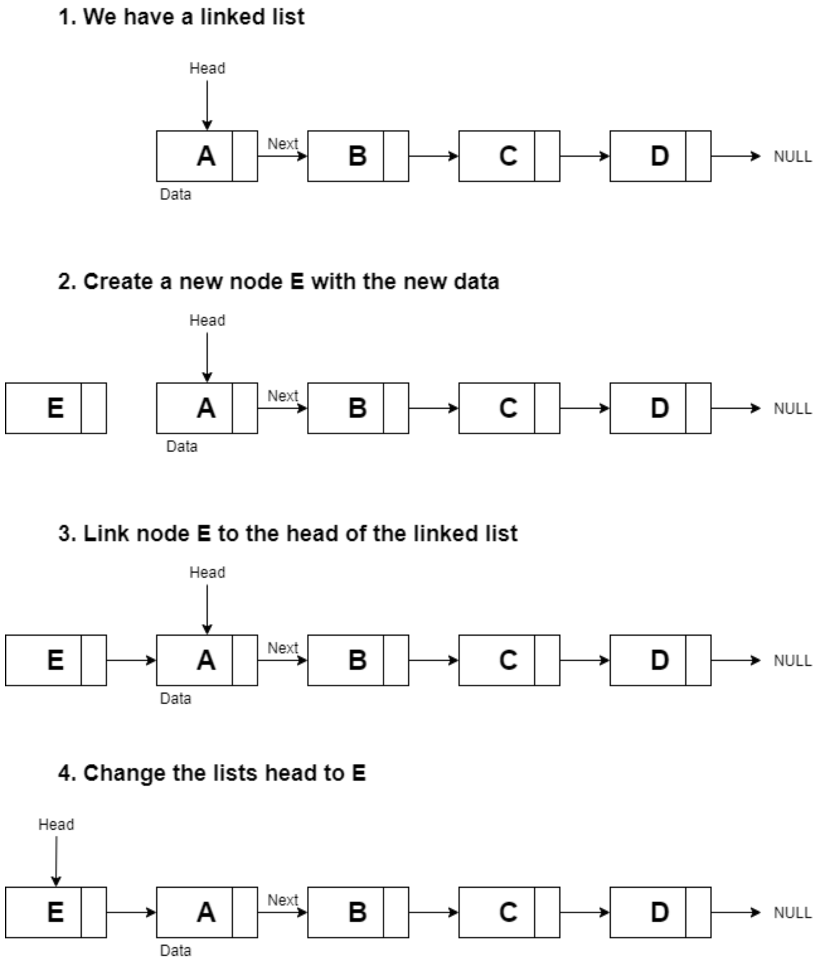
c) Create a Class called LinkedList. The Class should have one field called ‘head’. Head should be set to `None`. The head field represents the first element in the linked list. The Linked list should have a method: **`add(self, new_data`**`)`. This method creates a new node with the new data and places the node last in the chain.

Hint: check the following **Pseudo Code** for the add method:



```
def add(self, new_data):
    new_node =Node(new_data)
    if head does not exist:
        head = new_node
        stop
    last_node = head
    while last_node.next exist:
        last_node = last_node.next
    last_node.next = new_node
```

d) Implement another method **`add_at_beginning(self, new_data)`** which creates a node and places it at the beginning of the chain in the linked list. The following image contains a suggestion of what steps this method could take:



1. We have a linked list

2. Create a new node E with the new data

3. Link node E to the head of the linked list

4. Change the lists head to E