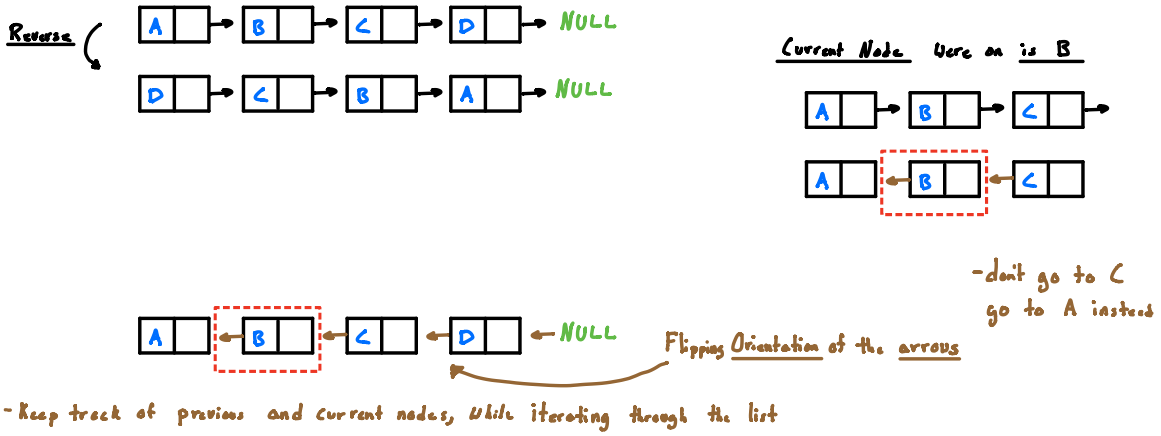
Linked List - Reverse List



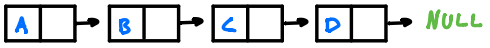

```
def reverse_iterative(self):

    previous = None
    current_node = self.head

    while current_node:

        next_node = current_node.next

        current_node.next = previous

        self.print_helper(previous, "Previous")
        self.print_helper(current_node, "Current Node")
        self.print_helper(next_node, "Next Node")
        print("\n")

    (A) previous = current_node

    (B) current_node = next_node

(C) self.head = previous
```

Keep track of Previous and Current Node

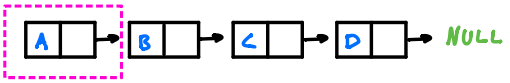
While current Node isn't equal to NONE, loop thru entire linked list

Step-by-Step


~ Current Node

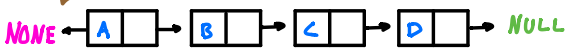


iteration 1

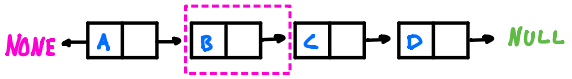


Previous : NONE
Current Node: A
Next Node : B

A Previous(NONE) = A
B Current Node = B ~ Changing node, continue loop

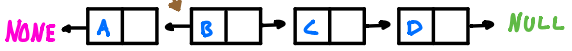


iteration 2

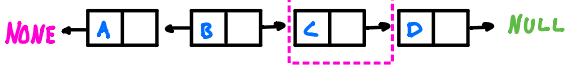


Previous : A
Current Node: B
Next Node : C

A Previous(A) = B
B Current Node = C ~ Changing node, continue loop



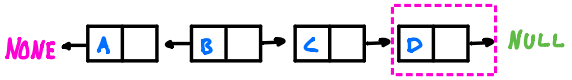
iteration 3



Previous : B
Current Node: C
Next Node : D

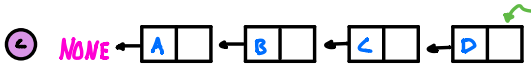
A Previous(B) = C
B Current Node = D ~ Changing node, continue loop

iteration 4



Previous : C
Current Node: D
Next Node : NONE

A Previous(C) = D
B Current Node = NONE ~ Exit loop



← last Step is to change the Previous (last node) to the head node.