Reverse a Linked List

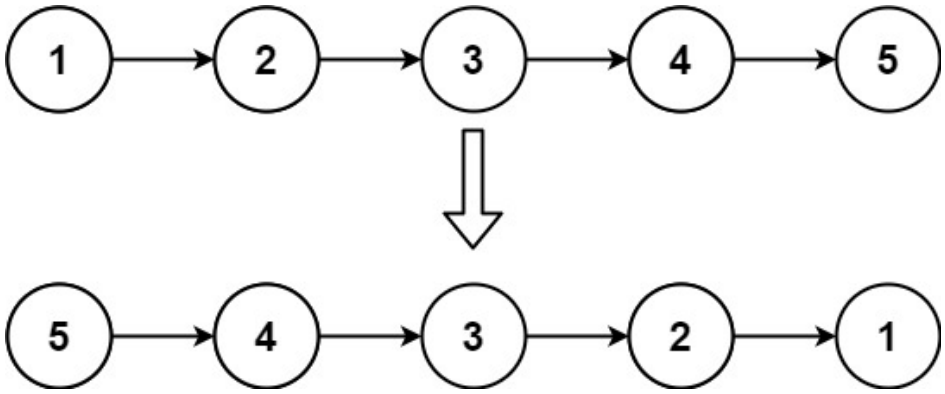
Difficulty	Easy
Category	LinkedList
Question	https://leetcode.com/problems/reverse-linked-list/
Solution	https://youtu.be/G0_I-ZF0S38
Status	Done

Question

> Given the `head` of a singly linked list, reverse the list, and return *the reversed list*.

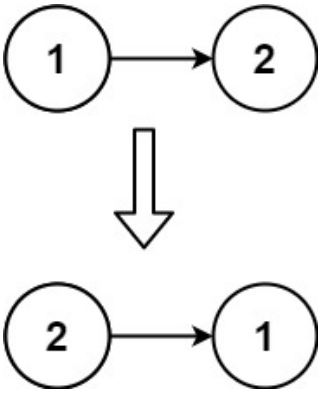
Example

Example 1:




```
Input: head = [1,2,3,4,5]
Output: [5,4,3,2,1]
```

Example 2:




```
Input: head = [1,2]
Output: [2,1]
```

Example 3:

```
Input: head = []
Output: []
```

Idea

💡 Use before, temp(Current Node), after to keep track of nodes and update accordingly

Solution

```
class Solution:
    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:
        # Initialize two pointers, 'before' and 'temp'.
        # 'before' will be used to reverse the direction of the links.
        # 'temp' is used for traversal.
        before, temp = None, head

        # Traverse the linked list.
        while temp:
            # Save the reference to the next node in 'after'.
            after = temp.next

            # Reverse the direction of the link by making the 'next' of the current node point to 'before'.
            temp.next = before

            # Move 'before' and 'temp' pointers one step forward.
            before = temp
            temp = after
```

```
        # 'before' will now point to the new head of the reversed linked list.
        return before
```