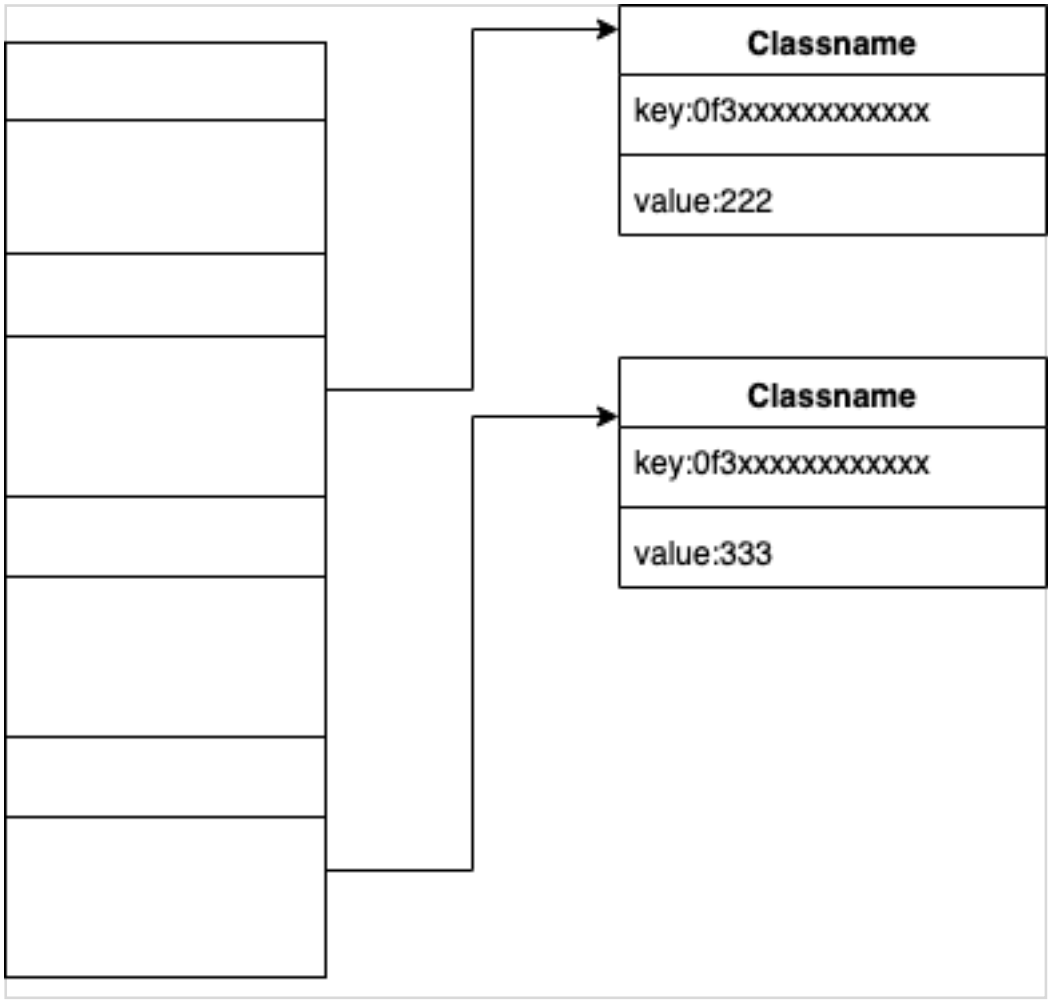
Leetcode 206.Reverse Linked List

链表本质是理解"**指针**"的概念。指针在各种语言都有运用到，只是叫法不同，就像Java中是"**引用**"。实际上，意思都是存储对象的内存地址。

> 将某个变量赋值给指针，实际上就是将这个变量的地址赋值给指针，或者反过来说，指针中存储了这个变量的内存地址，指向这个变量，通过内存地址可以找到这个变量值。

大体如下图：



下面通过一道简单的题目演示下链表移动。

Leetcode 206.Reverse Linked List

Reverse a singly linked list.

Example:

Input: 1->2->3->4->5->NULL

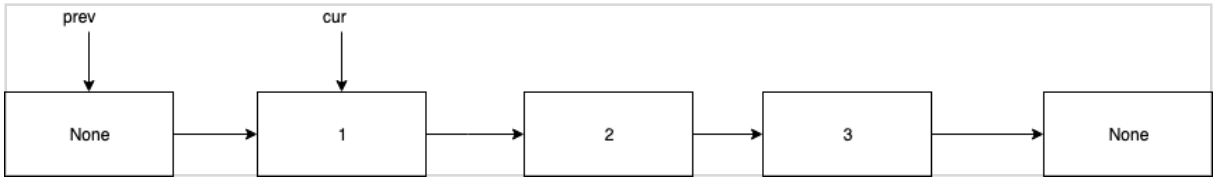
Output: 5->4->3->2->1->NULL

```
#author:kilien
```

```python
#Leetcode 206.Reverse Linked List
#思路：定义两个指针，指向前继节点和当前节点
#多元赋值，两两互换
#time:O(n)  space:O(1)
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, x):
#         self.val = x
#         self.next = None

class Solution:
    def reverseList(self, head: ListNode) -> ListNode:
        cur, prev = head, None
        while cur:
            cur.next, prev, cur = prev, cur, cur.next
        return prev
```

初始状态：

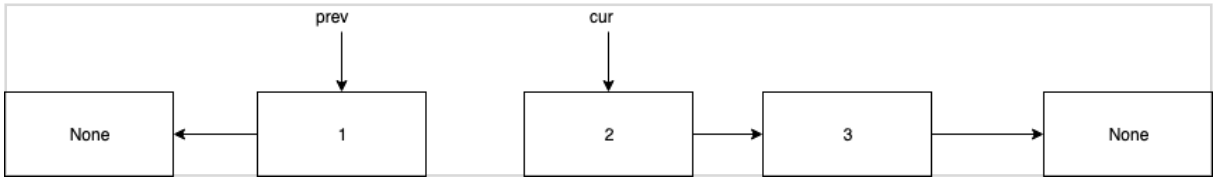


第一次循环：

将当前节点的后继指针指向前继节点，头节点的前继为**Null**；

将当前节点的前继指针指向当前节点，即指向本身，value为1；

当前节点指针赋值为当前节点的后继节点，即value为2的节点

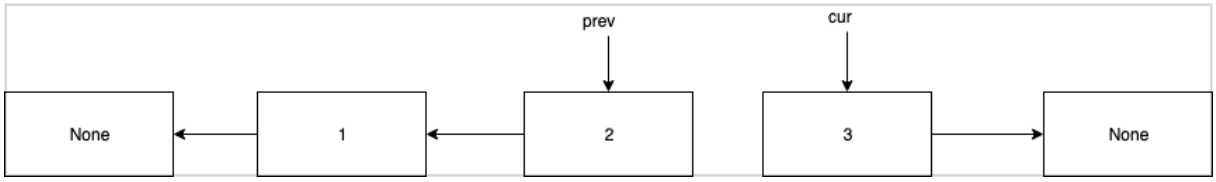


第二次循环：

将当前节点的后继指针指向前继节点，即value为1的节点；

将当前节点的前继指针指向当前节点，即value为2的节点；

当前节点指针赋值为当前节点的后继节点，即value为3的节点

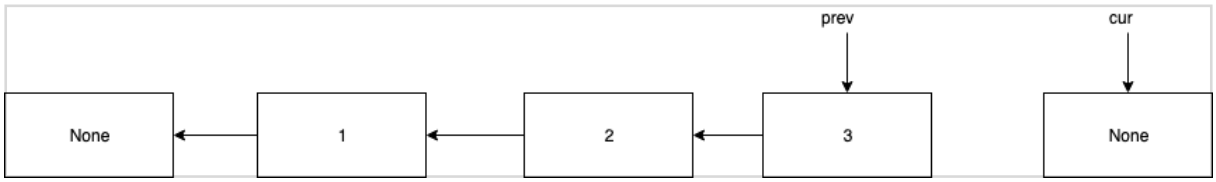


第三次循环：

将当前节点的后继指针指向前继节点，即value为2的节点；

将当前节点的前继指针指向当前节点，即value为3的节点；

当前节点指针赋值为当前节点的后继节点，即value为None的节点



注：python中多位赋值类似构建一个元组存储临时变量,如下两段代码等价

```
#cur.next, prev, cur = prev, cur, cur.next
tuples = (prev, cur, cur.next)
cur.next = tuples[0]
prev = tuples[1]
cur = tuples[2]
```

链表中使用该方式需要注意赋值顺序，错位将导致链表断裂

#algorithm#