Linked List Cycle

Question: Given a linked list, determine if it has a cycle in it

Solutions:

```
class ListNode:
    def __init__(self, x):
        self.val = x
        self.next = None
class Solution:
    # @param head, a ListNode
    # @return a boolean
    def hasCycle(self, head):
        if head == None or head.next == None:
            return False
        slow = fast = head
        while fast and fast.next:
            slow = slow.next
            fast = fast.next.next
            if slow == fast:
                return True
        return False


if __name__ == '__main__':
    ll, ll.next, ll.next.next = ListNode(2), ListNode(4), ListNode(3),
    ll.next.next.next = ll.next
    print( Solution().hasCycle(ll) )
```