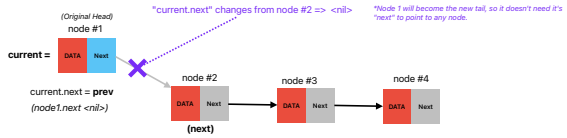
Linked List - ***reverse()***

*This document is a visual explanation for how Linked-List Reversal works in **GoLang**.*

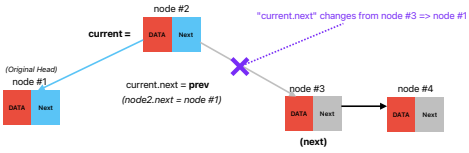
The visual diagrams below are based on the following code:

```
67
68  //! Reverse linked list
69  func (llist *LinkedList) Reverse() {
70      var prev, next *Node
71      current := llist.head
72
73      for current != nil {
74          next = current.next
75          current.next = prev
76          fmt.Printf("current.next: %v \n", current.next)
77          fmt.Printf("prev: %v \n", prev)
78          prev = current
79          current = next
80      }
81
82      llist.head = prev
83  }
84
```

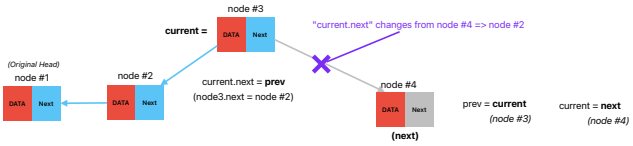
Loop, Iteration # 1



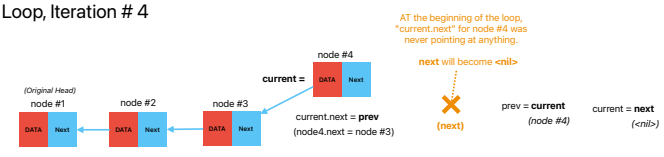
Loop, Iteration # 2



Loop, Iteration # 3



Loop, Iteration # 4



Loop - Break

During the attempt to start loop **iteration # 5**
- current (node #4's next) is <nil>

As a result, the loop will **break**
- This will move on to the next code, which will set the new head of the list to **prev**, which is node # 4.