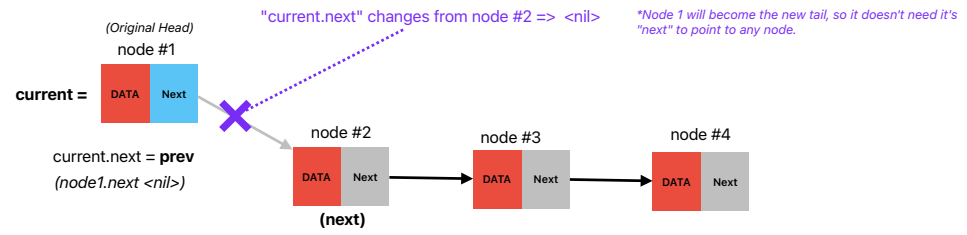
Linked List - ***reverse()***

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

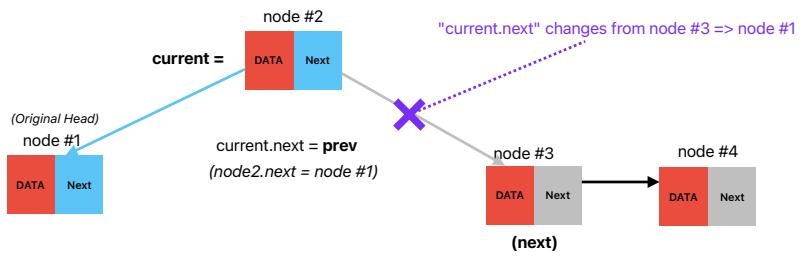
The visual diagrams below are based on the following code:

```go
//! Reverse linked list
func (llist *LinkedList) Reverse() {
    var prev, next *Node
    current := llist.head

    for current != nil {
        next = current.next
        current.next = prev
        fmt.Printf("current.next: %v \n", current.next)
        fmt.Printf("prev: %v \n", prev)
        prev = current
        current = next
    }

    llist.head = prev
}
```

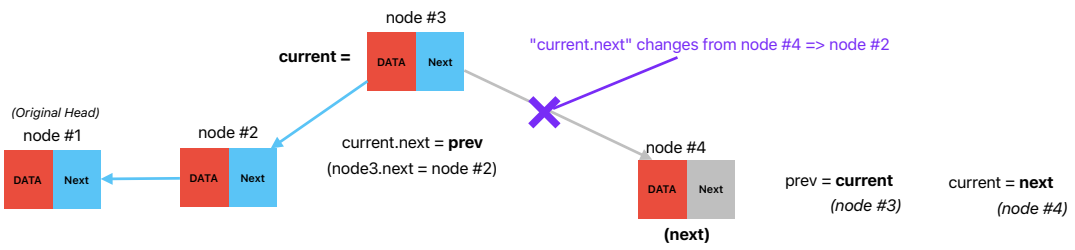
Loop, Iteration # 1



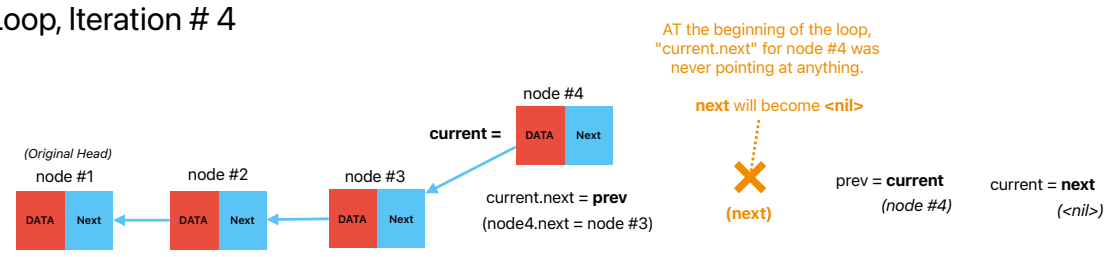
Loop, Iteration # 2



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.