Question: Given a node, Insert it at beginning of a doubly linked list.

Solution: To add a new Node at the beginning, check the head.

Algorithm:

If head is None:

simply make the next of new node to be None because we cannot add to the next or Previous of a node when that Node itself has nothing.

If head is a Node:

make Previous of head to be new node

make next of node to be the head

return new node, because it is the new node

Template to build a node

```
Class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None
```

```
def Insert(head, value):
    newNode = Node(value)
    // check if head is not None to make the connection for prev.
    if head is not None:
        head.prev = newNode
    // making new to be the new head by making its next old head
    newNode.next = head
    return head
```

Example

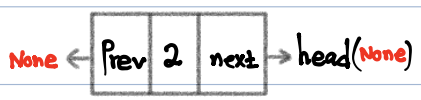
```
head = None
temp = Insert(head, 2)
// The value in temp is our new head
temp = Insert(temp, 4)
```

Step by step implementation

① head is None? yes — **Insert(head, 2)**

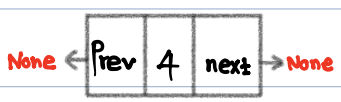
```
newNode = Node(2)
```


```
newNode.next = head
```

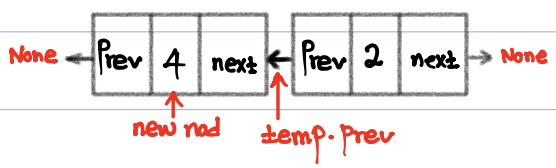


② head is None? No; **Insert(temp, 4):**

```
newNode = Node(4)
```




```
// head is not None, make its Prev → newNode
temp.prev = newNode
```



// make next of new node to point to old head

newNode.next = temp