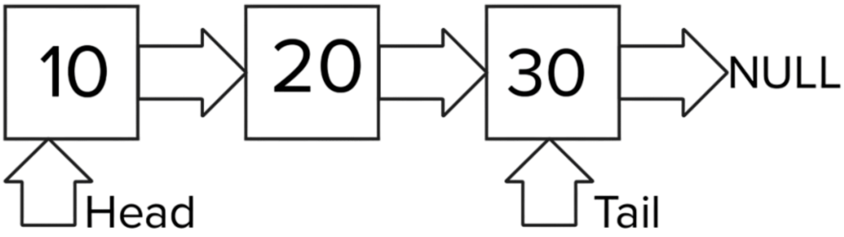
# 🧠 *A Visual Guide to Linked Lists*

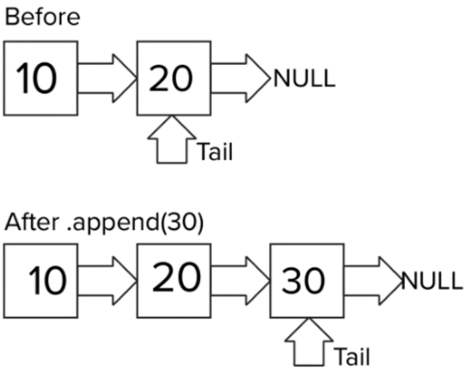
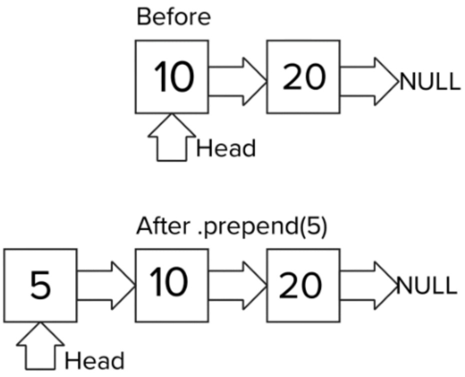
A **linked list** is a linear data structure composed of **nodes**. Unlike regular lists, its elements are not stored contiguously in memory, so each node contains a pointer to the next node's location.

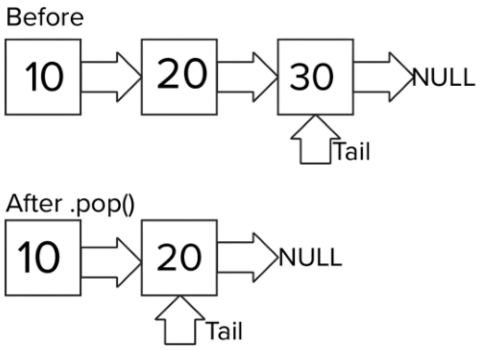
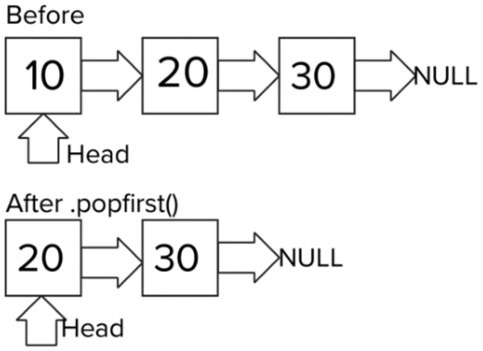
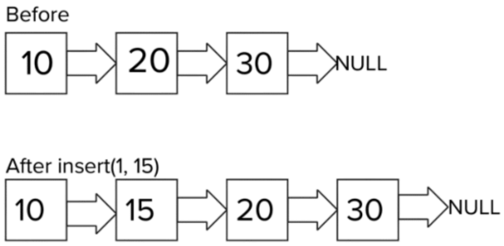
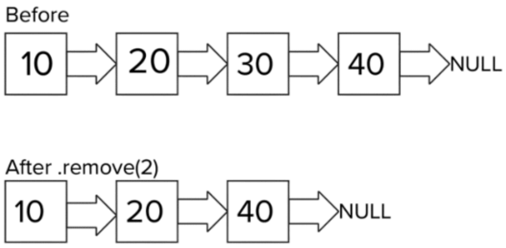
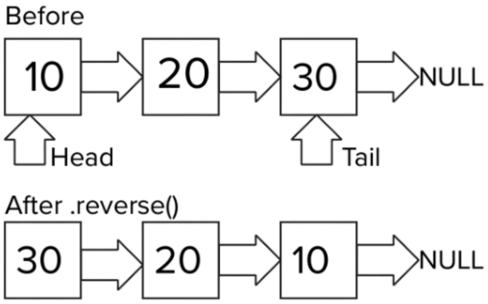
Each node contains two things:

- **Value**
- **Pointer** (to the next node)

🟢 The **head** is the first node
🔴 The **tail** is the last (points to None)



| Method | Visual | Time Complexity |
|---|---|---|
| **append** |  Before<br>10<br>20<br>NULL<br>Tail<br>After .append(30)<br>10<br>20<br>30<br>NULL<br>Tail | O(1) |
| **prepend** |  Before<br>10<br>20<br>NULL<br>Head<br>After .prepend(5)<br>5<br>10<br>20<br>NULL<br>Head | O(1) |

| Method | Visual | Time Complexity |
|---|---|---|
| pop |  Before<br>10<br>20<br>30<br>NULL<br>Tail<br>After .pop()<br>10<br>20<br>NULL<br>Tail | O(n) |
| popfirst |  Before<br>10<br>20<br>30<br>NULL<br>Head<br>After .popfirst()<br>20<br>30<br>NULL<br>Head | O(1) |
| insert |  Before<br>10<br>20<br>30<br>NULL<br>After insert(1, 15)<br>10<br>15<br>20<br>30<br>NULL | O(n) |
| remove |  Before<br>10<br>20<br>30<br>40<br>NULL<br>After .remove(2)<br>10<br>20<br>40<br>NULL | O(n) |
| reverse |  Before<br>10<br>20<br>30<br>NULL<br>Head<br>Tail<br>After .reverse()<br>30<br>20<br>10<br>NULL | O(n) |