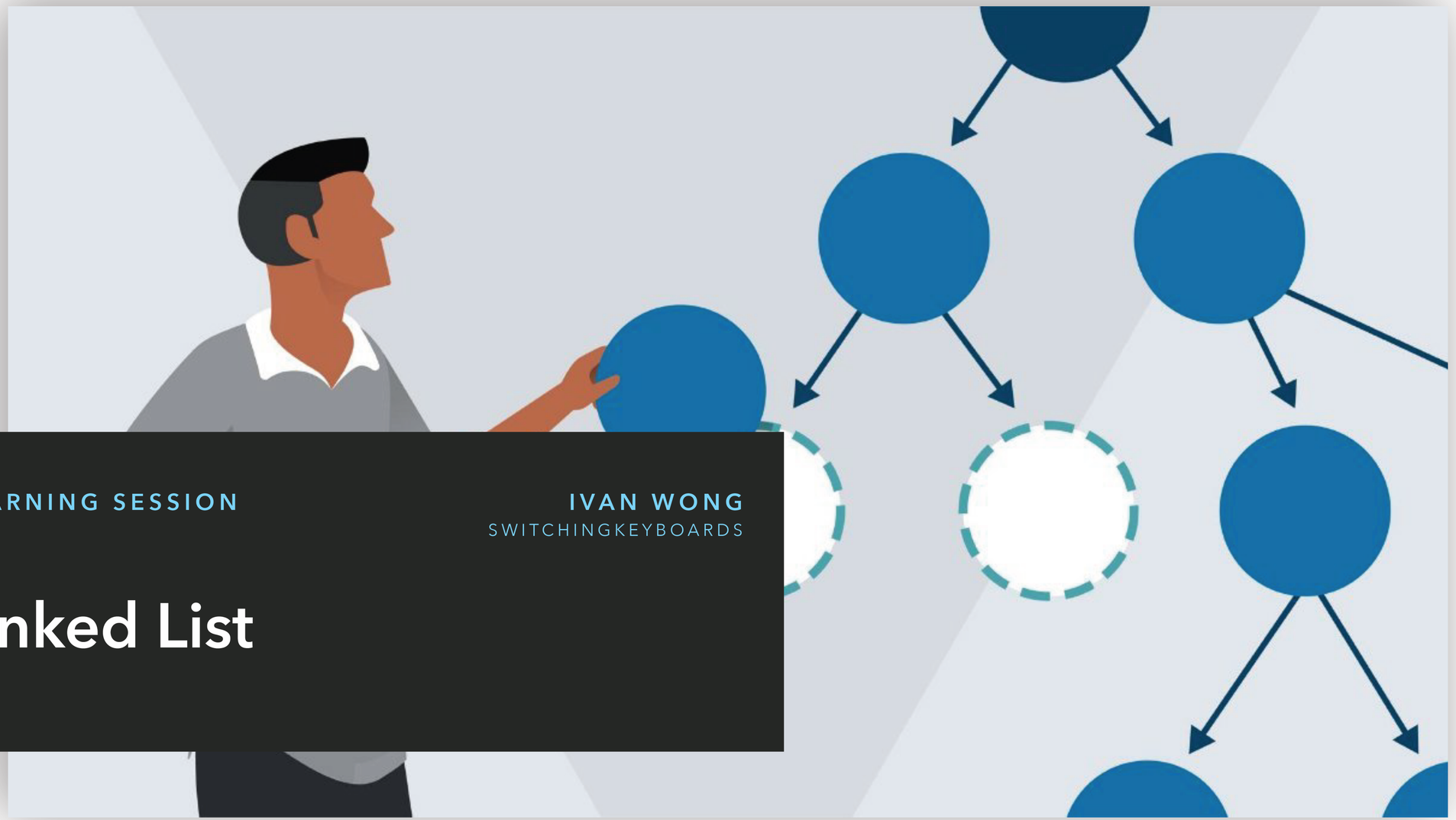


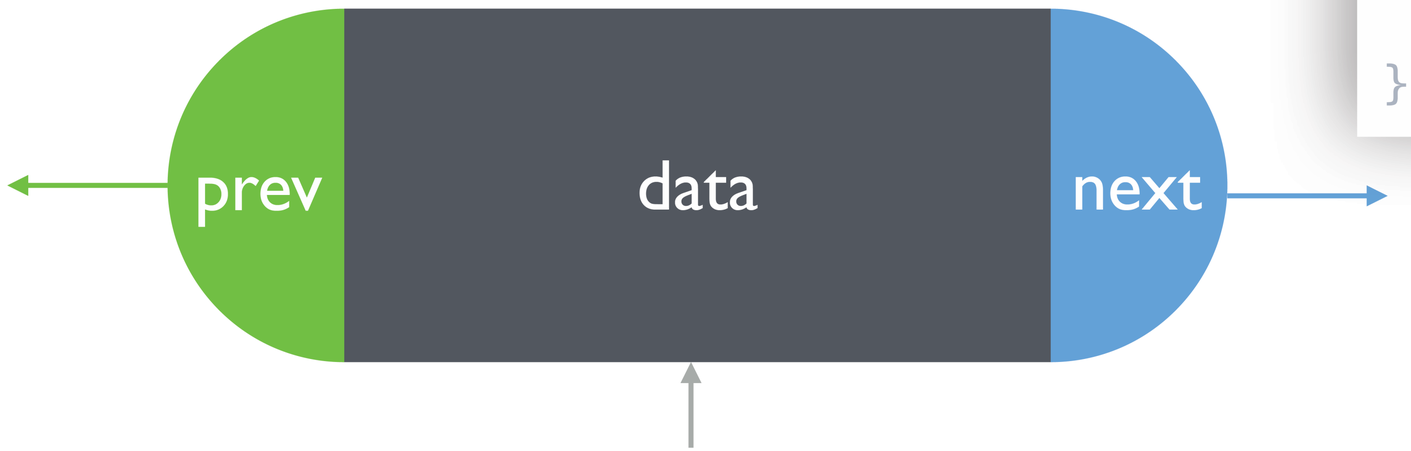
LEARNING SESSION

IVAN WONG
SWITCHINGKEYBOARDS

Linked List

What is a node?

```
class Node {
  constructor(data) {
    this.prev = null;
    this.data = data;
    this.next = null;
  }
}
```



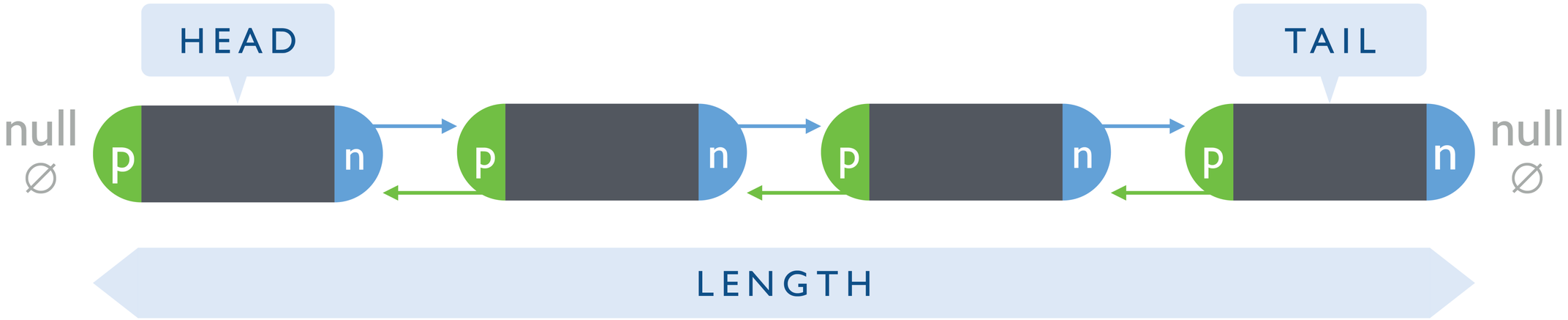
link to the previous node

data is stored here

link to the next node

Note: This is the node structure for a doubly-linked list, nodes in a singly-linked list have no head.

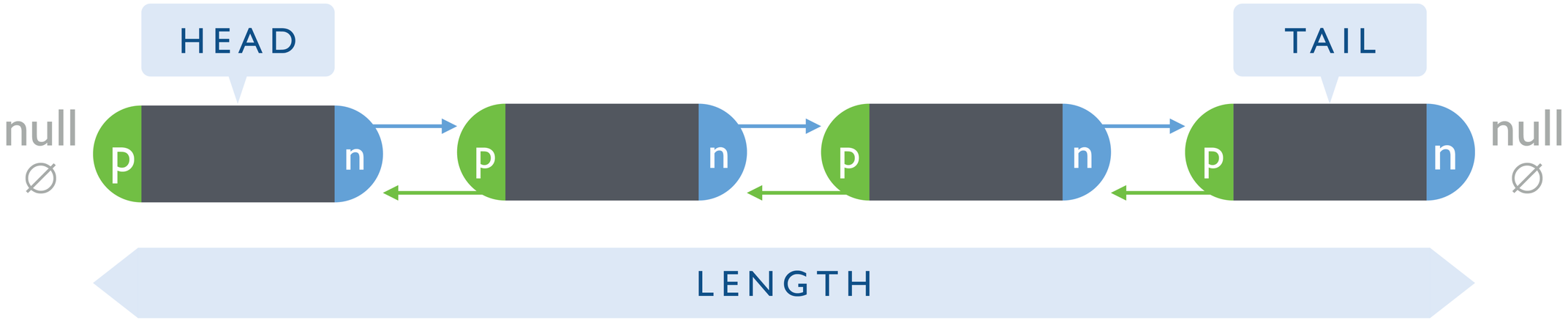
A linked list is like an array of nodes



A linked list is like an array of nodes

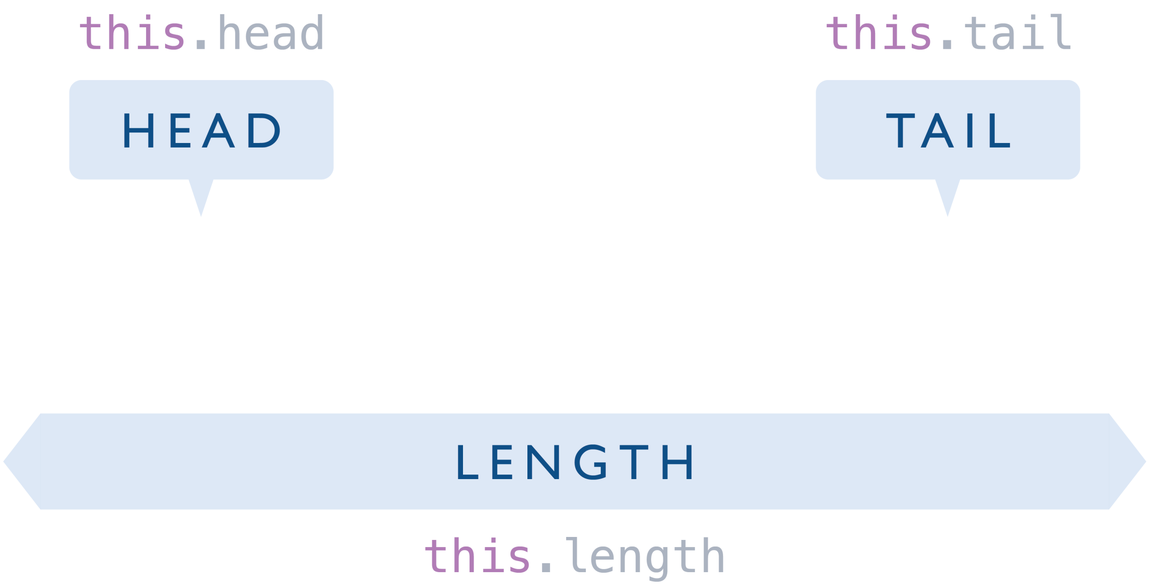
`push()` *(insert value at back)*
`pop()` *(remove value at back)*
`shift()` *(remove value at front)*
`unshift()` *(insert value at front)*

`delete()` *(remove specific value)*
`count()` *(keep track of list length)*



Constructor function

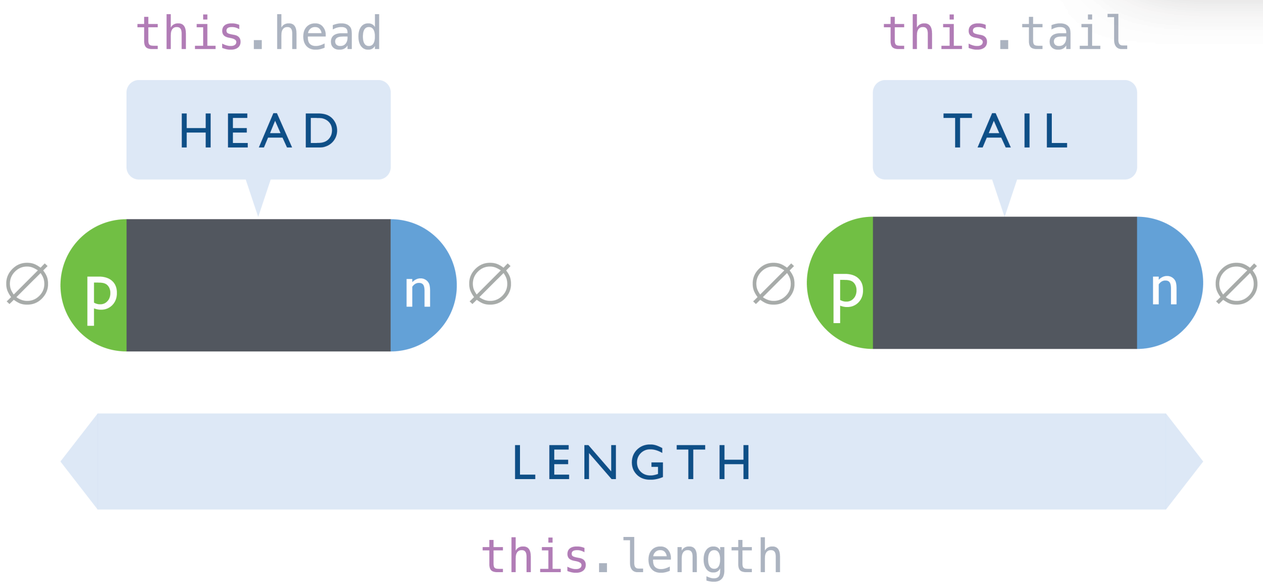
```
class LinkedList {
 constructor() {
    this.head = null;
    this.tail = null;
    this.length = 0;
 }
}
```



push() method

Creating the initial head and tail nodes

```
push(data) {
    const node = new Node(data);
    if (this.head === null) {
      this.head = node;
    }
    this.tail = node;
    this.length ++;
  }
```

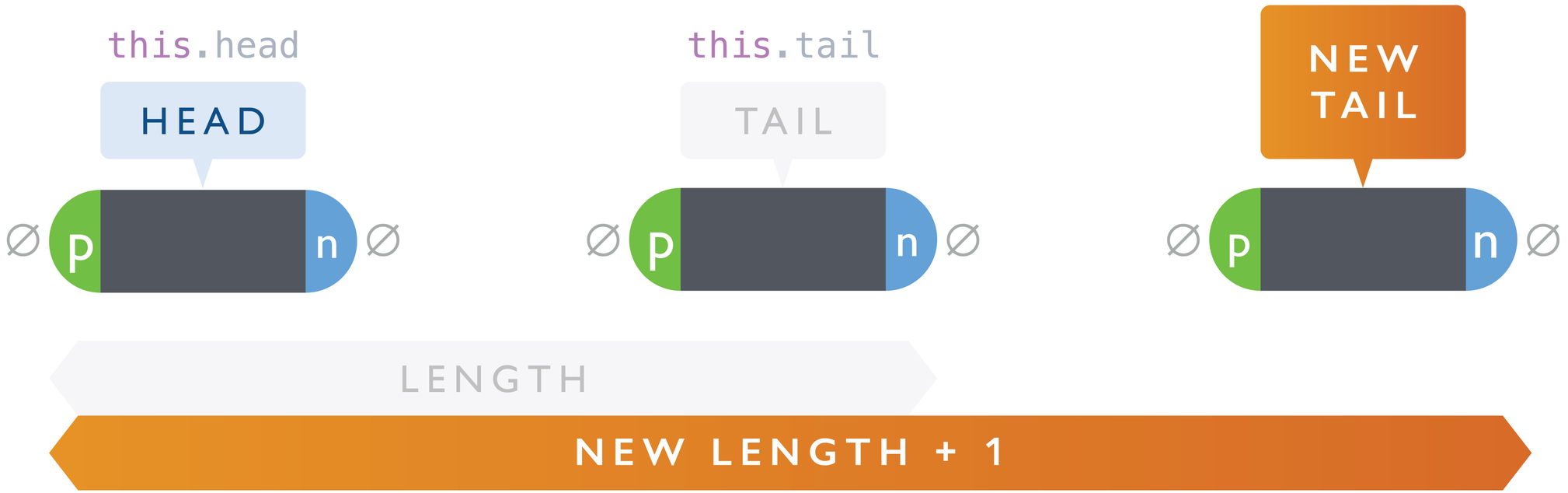


push() method

Thereafter…

```
this.head = node;
    } else {
      node.prev = this.tail;
      this.tail.next = node;
    }
this.tail = node;
```

```
const node = new Node(data);
```

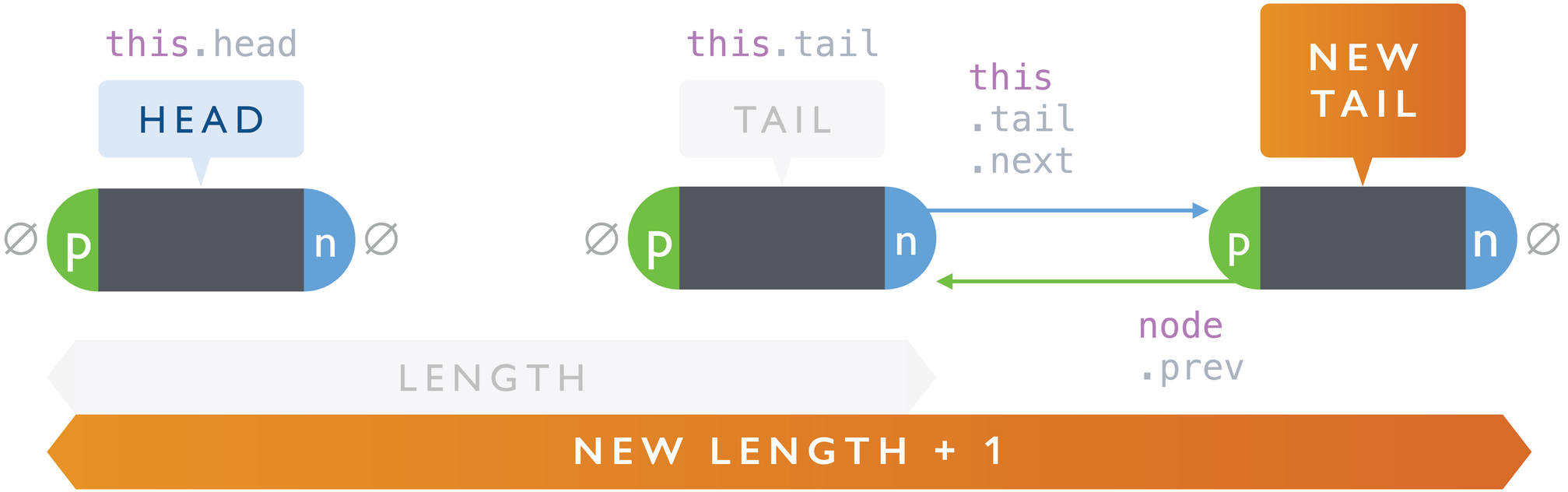


push() method

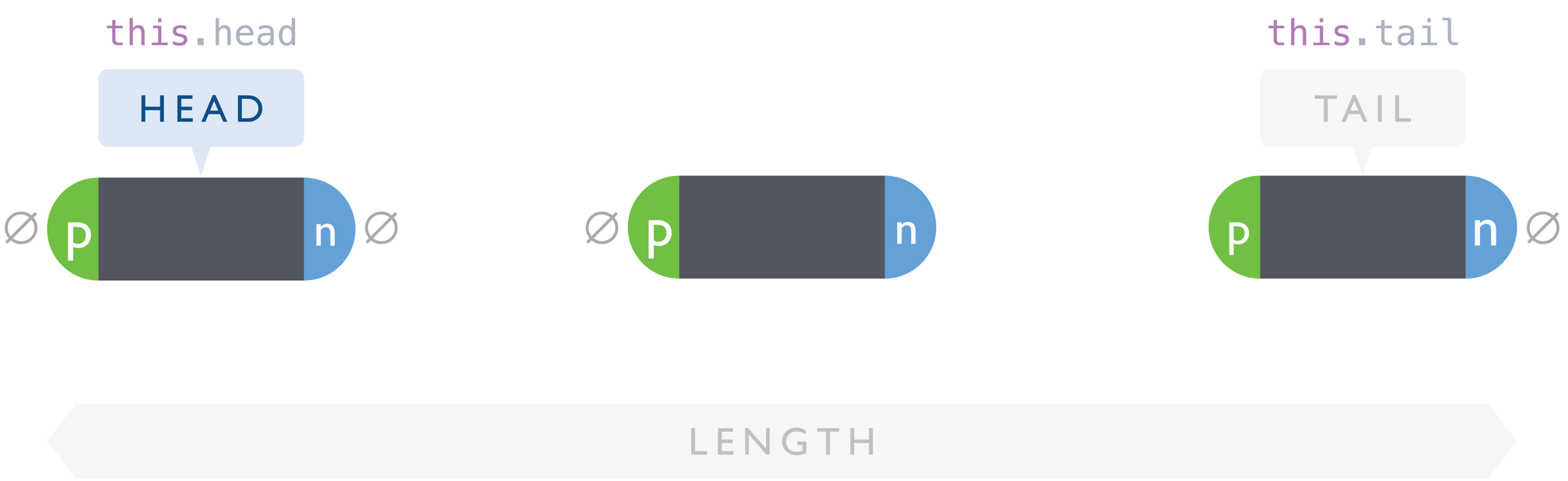
Thereafter…

```
this.head = node;
    } else {
      node.prev = this.tail;
      this.tail.next = node;
    }
this.tail = node;
```

```
const node = new Node(data);
```



pop() method

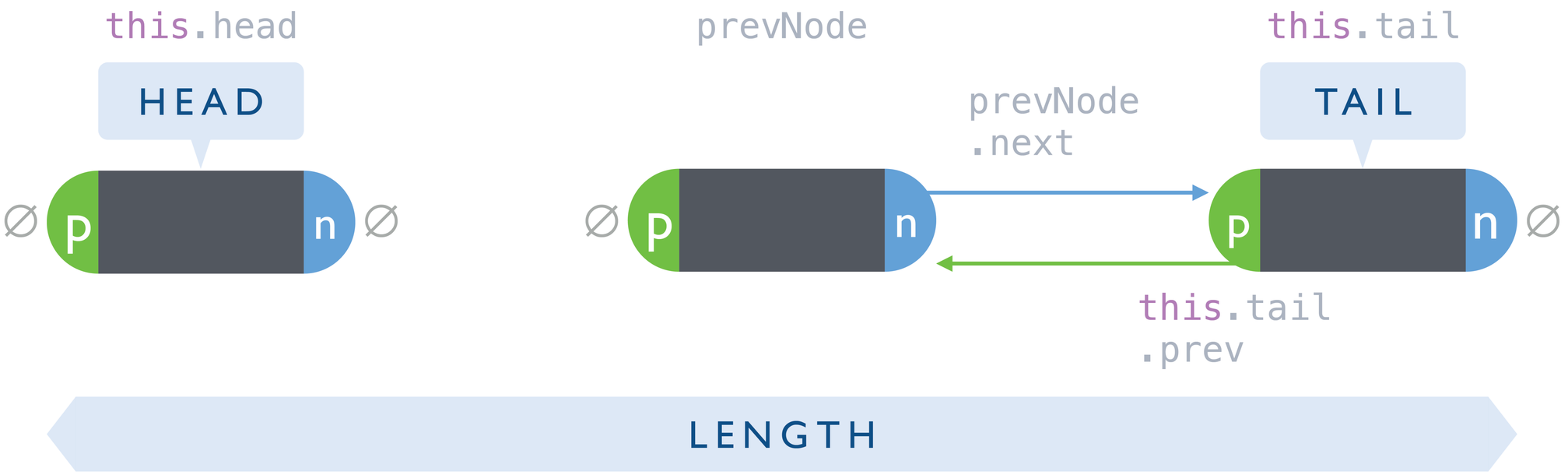


pop() method

If there is a previous node

```
const prevNode =
this.tail.prev;

if (prevNode !== null) {
  prevNode.next = null;
  this.tail = prevNode;
}
```

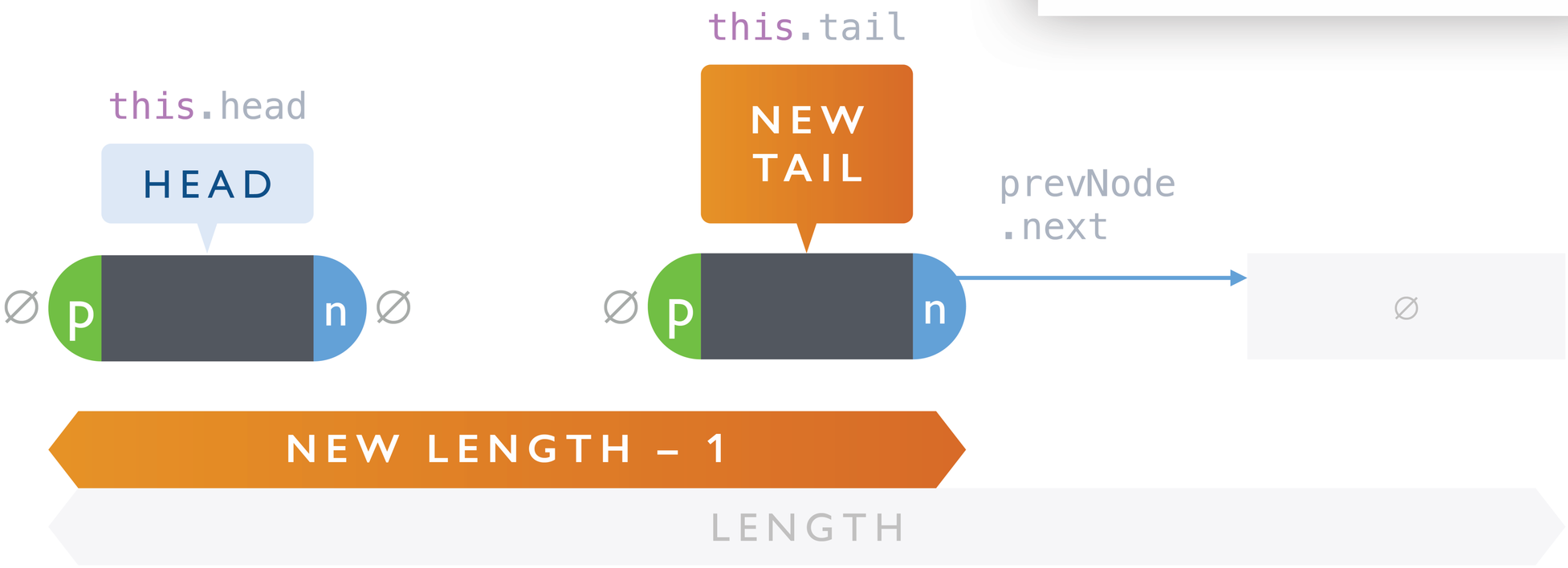


pop() method

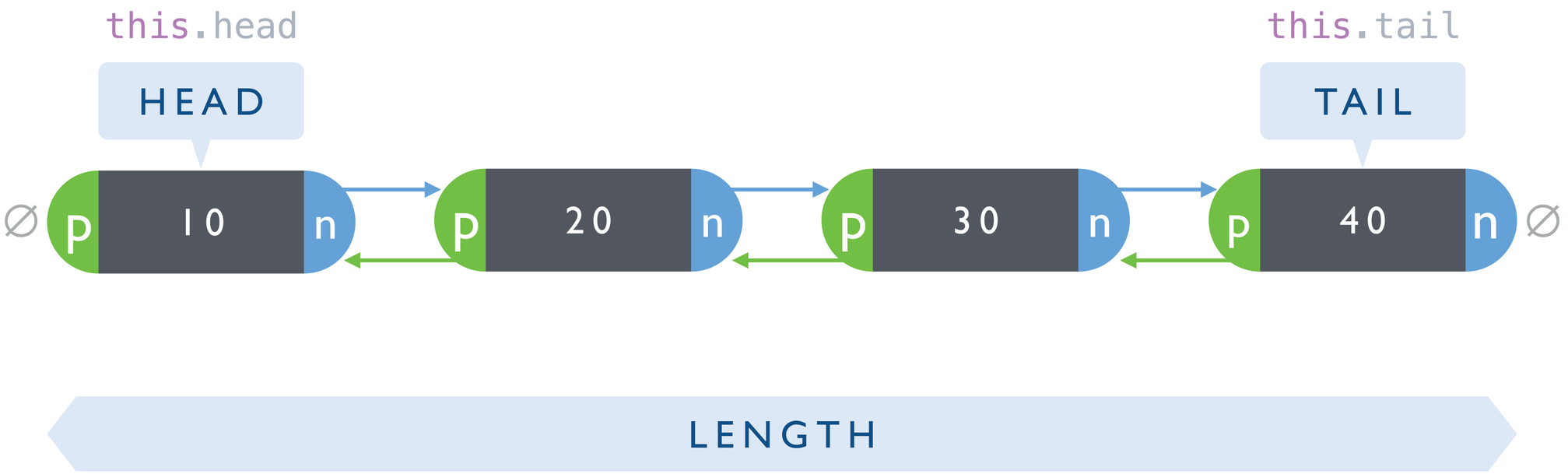
If there is a previous node

```
const prevNode =
this.tail.prev;

if (prevNode !== null) {
  prevNode.next = null;
  this.tail = prevNode;
}
```



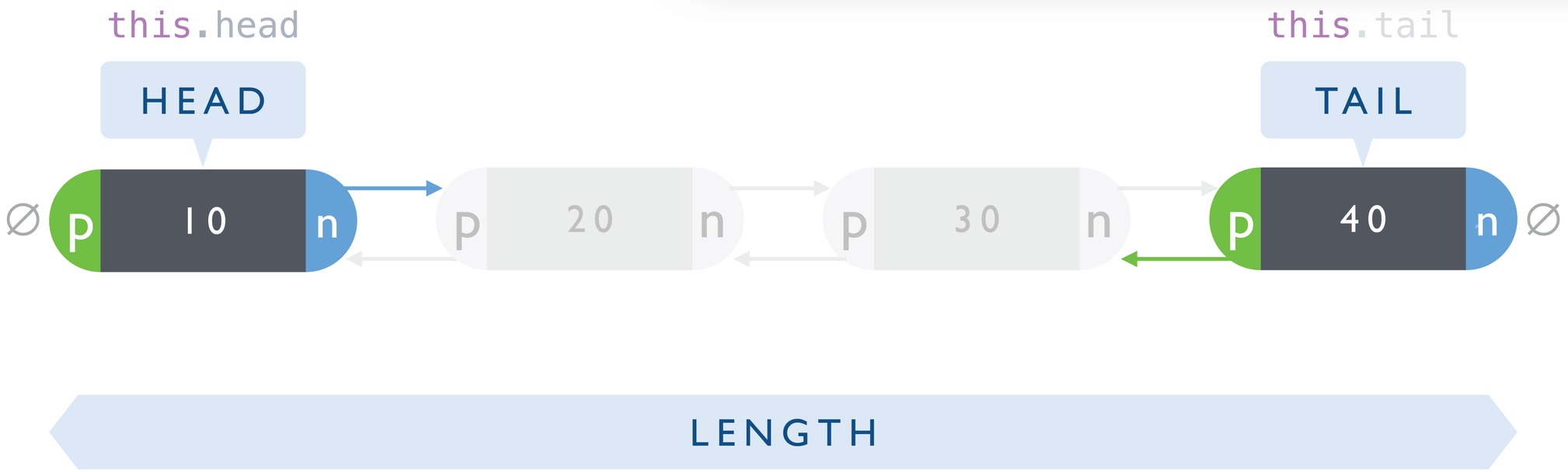
delete() method



delete() method

```
delete(10);
delete(40);
```

```
delete(data) {
  let nextNode = this.head.next;
  if (this.head.data === data) {
    this.shift();
  } else if (this.tail.data === data) {
    this.pop();
  }
```



delete() method

```
delete(20);
delete(30);
delete(50);
```

```
    this.pop();
    } else if (nextNode === null) {
      return;
    } else {
      while (nextNode.data !== data) {
        nextNode = nextNode.next;
      }
      nextNode = null;
      this.length--;
    }
  }
```