

Singly vs Doubly Linked List - JavaScript Guide

1. Difference Between Array, Singly, and Doubly Linked List

Array:

- Built-in structure in JavaScript.
- Fixed continuous memory blocks.
- Uses index to access elements.

Singly Linked List:

- Made using custom Node class.
- Each node points to the next node only.
- Memory can be scattered.
- One-directional traversal.

Doubly Linked List:

- Each node points to both next and previous.
- Two-directional traversal.
- More memory due to extra pointer.

2. Singly Linked List Code in JavaScript

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

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

  push(value) {
    const newNode = new Node(value);
    if (!this.head) {
```

Singly vs Doubly Linked List - JavaScript Guide

```
        this.head = newNode;
        this.tail = newNode;
    } else {
        this.tail.next = newNode;
        this.tail = newNode;
    }
    this.length++;
}

print() {
    let current = this.head;
    while (current) {
        console.log(current.value);
        current = current.next;
    }
}

const list = new SinglyLinkedList();
list.push("A");
list.push("B");
list.push("C");
list.print();
```

3. Doubly Linked List Code in JavaScript

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

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

Singly vs Doubly Linked List - JavaScript Guide

```
push(value) {
  const newNode = new Node(value);
  if (!this.head) {
    this.head = newNode;
    this.tail = newNode;
  } else {
    this.tail.next = newNode;
    newNode.prev = this.tail;
    this.tail = newNode;
  }
  this.length++;
}

printForward() {
  let current = this.head;
  while (current) {
    console.log(current.value);
    current = current.next;
  }
}

printBackward() {
  let current = this.tail;
  while (current) {
    console.log(current.value);
    current = current.prev;
  }
}

const dlist = new DoublyLinkedList();
dlist.push("A");
dlist.push("B");
dlist.push("C");
dlist.printForward();
dlist.printBackward();
```