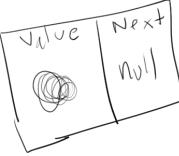
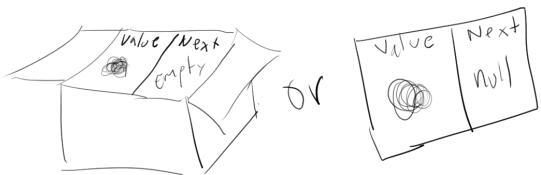


```
class Node {  
    constructor(value) {  
        this.value = value  
        this.next = null  
    }  
}
```



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



let n1 = new Node(3) → [3 | null]

let n2 = new Node(5) → [5 | null]

let n3 = new Node(1) → [1 | null]

let n4 = new Node(6) → [6 | null]

```

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

```



let n1 = new Node(3) → [3 | null]

let n2 = new Node(5) → [5 | null]

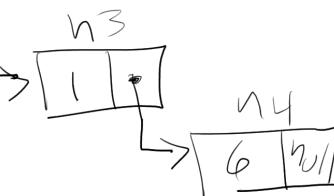
let n3 = new Node(1) → [1 | null]

let n4 = new Node(6) → [6 | null]

n1.next = n2



n2.next = n3



n3.next = n4

→ visualized



```

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

```



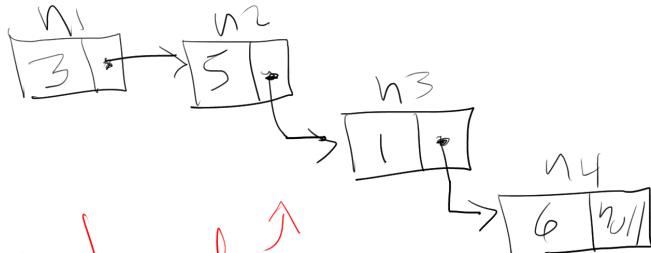
```
let n1 = new Node(3) → [3 | null]
```

```
let n2 = new Node(5) → [5 | null]
```

```
let n3 = new Node(1) → [1 | null]
```

```
let n4 = new Node(6) → [6 | null]
```

```
n1.next = n2
```



```

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

```



```

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

```



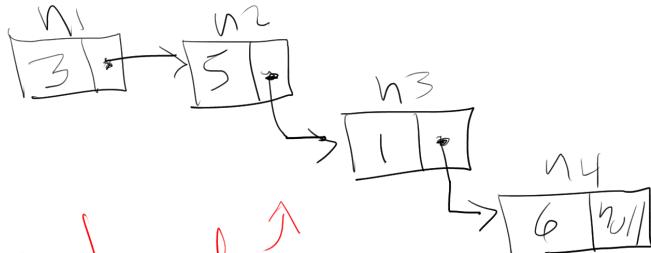
let n1 = new Node(3) → [3 | null]

let n2 = new Node(5) → [5 | null]

let n3 = new Node(1) → [1 | null]

let n4 = new Node(6) → [6 | null]

n1.next = n2



n2.next = n3

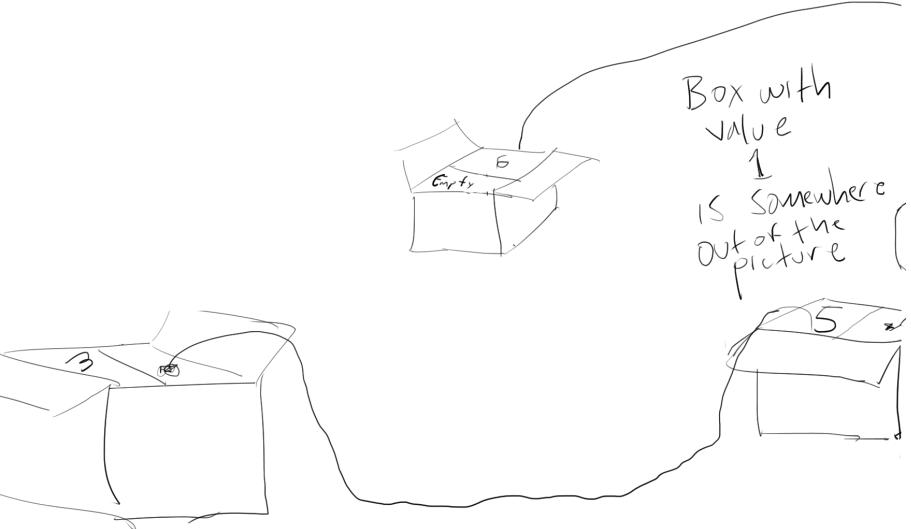
n3.next = n4

→ visualized

```

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

```



```

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

```



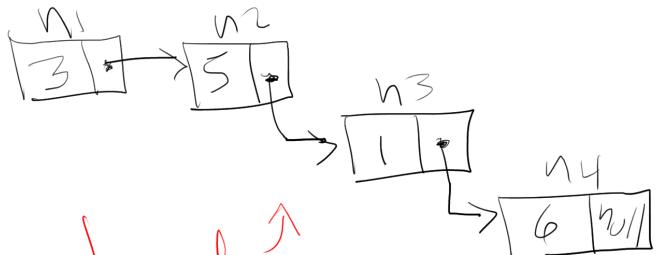
```
let n1 = new Node(3) → [3 | null]
```

```
let n2 = new Node(5) → [5 | null]
```

```
let n3 = new Node(1) → [1 | null]
```

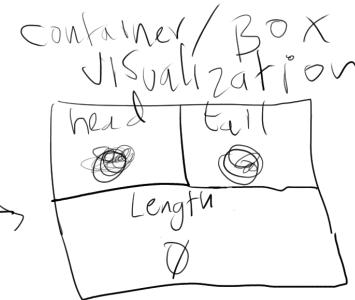
```
let n4 = new Node(6) → [6 | null]
```

```
n1.next = n2
```



```
class LinkedList {
```

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



```
addFirst(val) { ... }
```

```
addLast(val) { ... }
```

```
addAtIndex(index, val) { ... }
```

```
contains(val) { ... }
```

```
// And other methods
```

Why a LinkedList class?

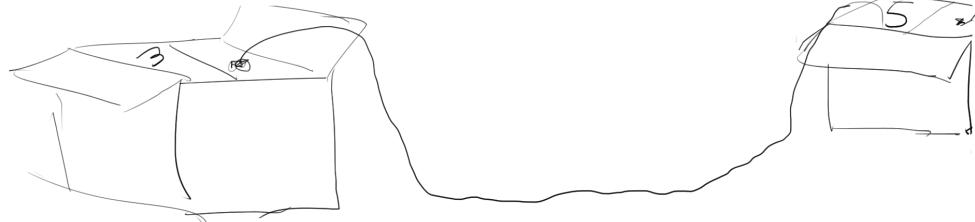
★ Ease of use

- Keeping track of key info
 - Head / tail
 - length

• Methods

Box with
value

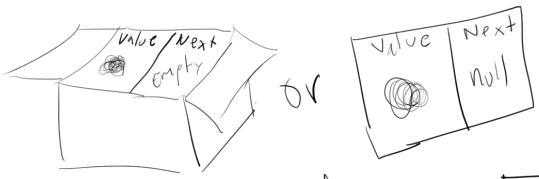
1
is somewhere
out of the
picture



```

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

```



```
let n1 = new Node(3) → [3 | null]
```

```
let n2 = new Node(5) → [5 | null]
```

```
let n3 = new Node(1) → [1 | null]
```

```
let n4 = new Node(6) → [6 | null]
```

```
n1.next = n2
```



```
n2.next = n3
```



```
n3.next = n4
```



→ visualized

```
class LinkedList {
```

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

```
addFirst(val) { ... }
```

```
addLast(val) { ... }
```

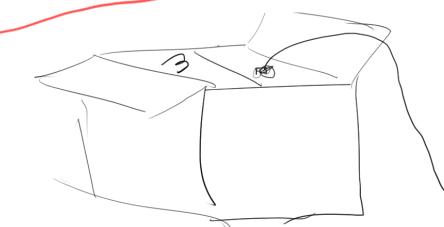
```
addAtIndex(index, val) { ... }
```

```
contains(val) { ... }
```

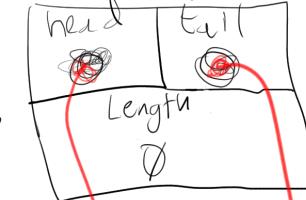
```
// And other methods
```

```
}
```

(at $ll = \text{new LinkedList}()$)
 $ll.addFirst(3)$



Container / Box
Visualization



Why a LinkedList
class?

★ Ease of use

- Keeping track of key info
 - Head / tail
 - length

• Methods

Box with
value

1
is somewhere
out of the
picture



```

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

```



let n1 = new Node(3) →

3	null
---	------

let n2 = new Node(5) →

5	null
---	------

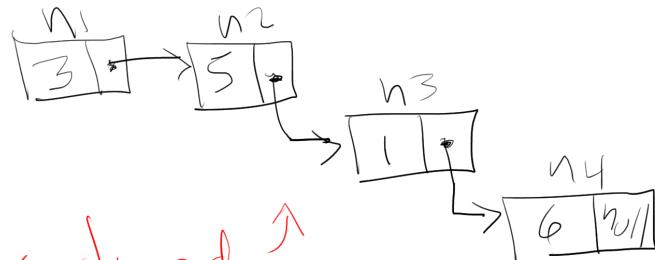
let n3 = new Node(1) →

1	null
---	------

let n4 = new Node(6) →

6	null
---	------

n1.next = n2



n2.next = n3

n3.next = n4

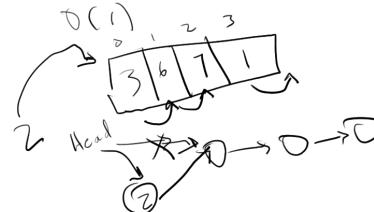
What do you remember?

• Pointers

- To find you have to \star traverse through Head

• Advantages

- removal / add at begin/end



• Disadvantage

- $O(1)$ lookup

Abstract Data Structure structured way of storing info

- Simple outside - complex inside

- Conceptual \star

- Few ways to implement

- Not Built-in \star

Box with
value