

THE INTRODUCTION TO

THE LINKED LIST

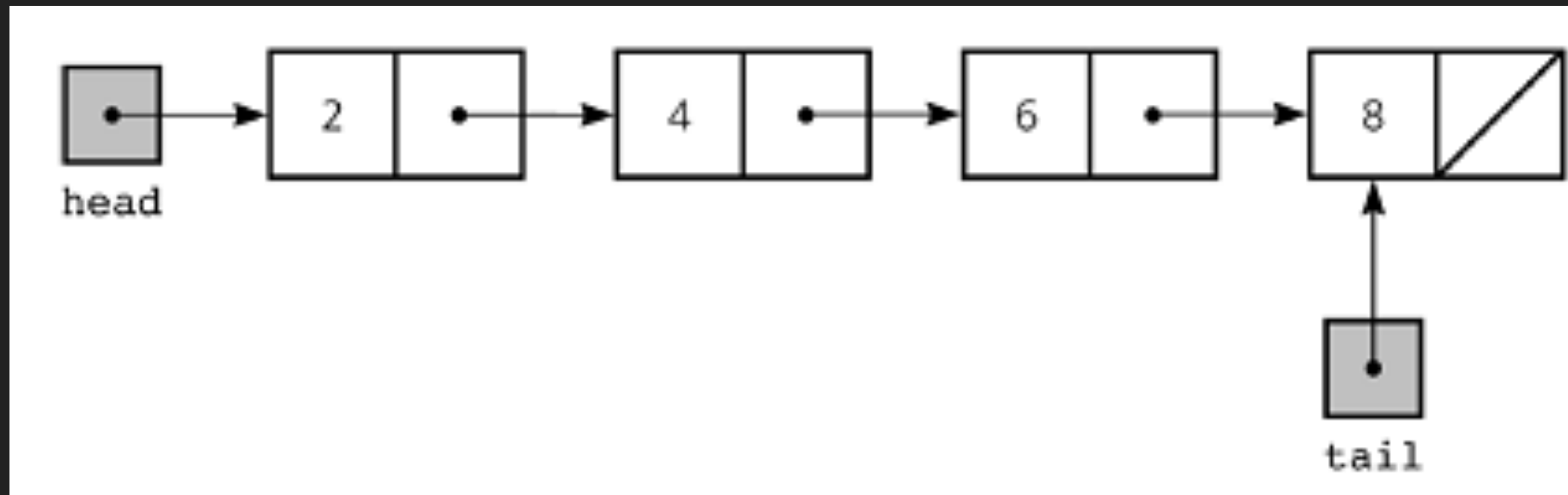
▶ What is a Data Structure??

- ▶ A data structure is a particular way of organizing data in a computer so that it can be used effectively.
- ▶ In computer science, a data structure is a data organization, management, and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operation that can be applied to the data. [Wiki](#)

▶ What is a Linked list???

- ▶ A linked list is a data structure that stores multiple values in a linear fashion. Each value in a linked list is contained in its own node, an object that contains the data along with a link to the next node in the list. The link is a pointer to another node object or null if there is no next node.

DATA STRUCTURE - LINKED LIST



- ▶ Like arrays, Linked Lists store data elements in sequential order. Instead of keeping indexes, linked lists hold pointers to other elements. The first node is called the head - while the last node is called the tail.

▶ The Design:

- ▶ First we need to set up the node structure.
- ▶ Each node contains data and a pointer to the next node in the list.

```
5  class Node {  
6      constructor(data, next = null) {  
7          this.data = data;  
8          this.next = next;  
9      }  
10 }
```

- ▶ In the Node class, the data property contains the value and the linked list item stores, and the next property is the pointer to the next item in the list. The next property starts out as null - because we don't know what the next node is.

- ▶ Now we need to add the LinkedList Class.

```
12  ✓ class LinkedList {
13  ✓     constructor() {
14      |     this.head = null;
15      |     }
16  | } You, a few seconds
```


- ▶ The LinkedList class represents a linked list and will contain methods for interacting with the data it contains.
- ▶ The first method we will code will be when we need to add new data to the list

- ▶ Adding an item to our linked list required us walking the structure to find the correct location, creating a new node, and inserting it in place. Remember!! There is the case when the list is empty, in which you simply create a new node and assign it to head.

DATA STRUCTURE - LINKED LIST

```
17  add(data){
18      //create a new node
19      const newNode = new Node(data);
20      //handel the case when no items are in the list yet
21      if(this.head === null) {
22          this.head = newNode;
23      } else {
24          let current = this.head;
25          //now loop through the list until we reach the end
26          while(current.next !== null) {
27              current = current.next
28          }
29          //now make the node into the next pointer
30          current.next = newNode;
31      }
32  }
```

You, a few seconds ago

- ▶ Now we need to find a value at a specific spot.
- ▶ We need to do this by referencing an index. This index will show us what the value is at a specific number in the list.

DATA STRUCTURE - LINKED LIST

```
getAt(index) {  
  let counter = 0;  
  let node = this.head;  
  while (node) {  
    if (counter === index) {  
      return node;  
   }  
  
    counter++;  
    node = node.next;  
  }  
  return null;  
}
```

- ▶ Now it's up to you to add multiple methods for our list.
- ▶ What I want to be able to do is:
 - Remove the first node in the list.
 - Remove the last node in the list.
 - Remove a node at a specified index.
 - Insert a node at a specified index.
 - Clear the list completely.

- ▶ Remember that there will be some edge cases for some of these methods:

The list is empty (no traversal is possible)

The index is less than zero

The index is greater than the number of items in the list

The index is zero (removing the head)