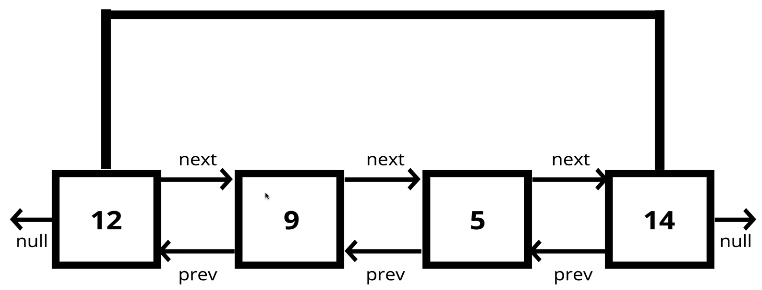
JavaScript Algorithms and Data Structures Masterclass

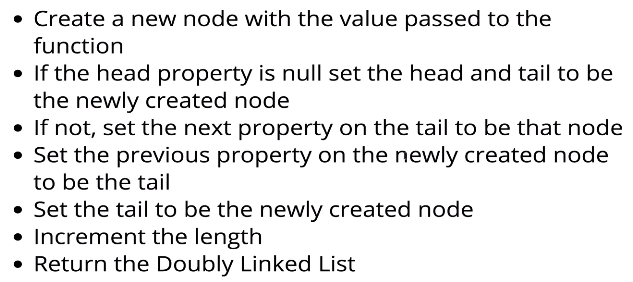
# Section 20: Doubly Linked Lists

## What is it?

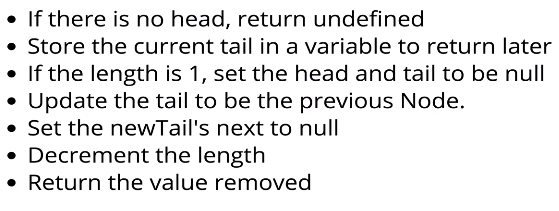
* Almost identical to SLL, but each node has .next and .previous
  + **Diagram**:
    - 
* Comparison with SLL
  + More memory for More Flexibility (trade off)

## Methods:

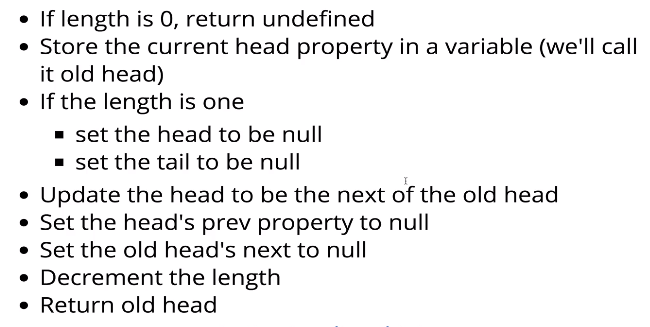
### Push()

* Add node to the **end** of the DLL
  + 

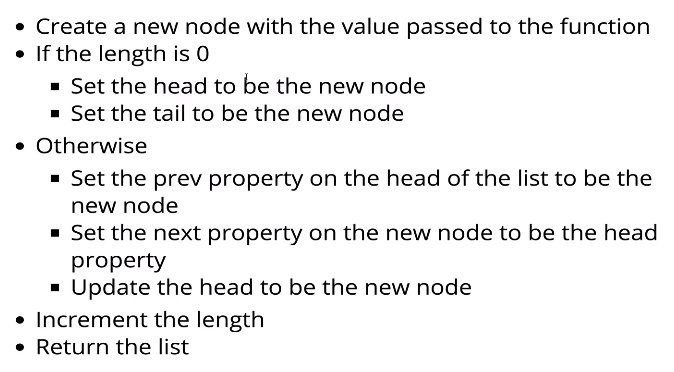
### Pop()

* Remove node from **end** of DLL
  + **Psuedo-code**:
    - 
      * Also, make the old tail’s previous null too!

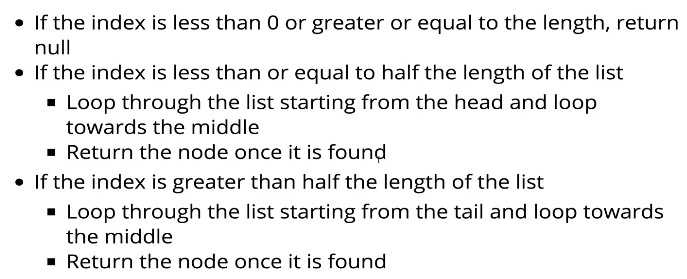
### Shift()

* Remove the node at the **beginning** of the DLL
  + **Psuedo-code**:
    - 

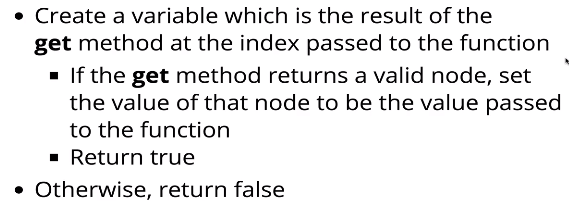
### Unshift()

* Add a node to the **beginning** of the DLL
  + 

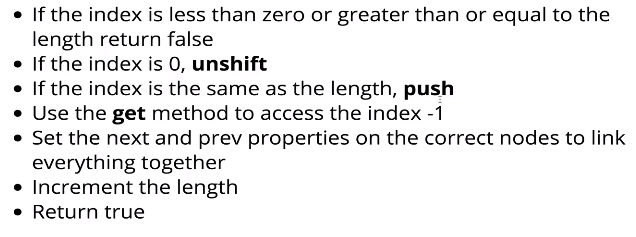
### Get()

* Access a node in DLL at its index position
  + 

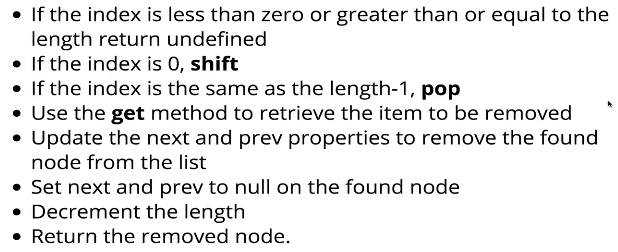
### Set()

* **Replace** the value of a node in the DLL at an index
  + 

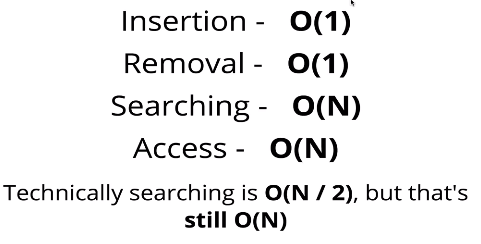
### Insert()

* Add a node into DLL at a certain index
  + 

### Remove()

* Remove a node in the DLL at a certain index
  + 

## Big O of DLLs

* 
  + Searching is *divide and conquer* since we search from either sides of the DS and move inwards
* **Recap**:
  + 