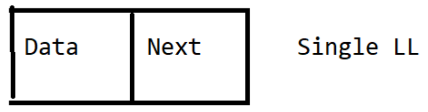
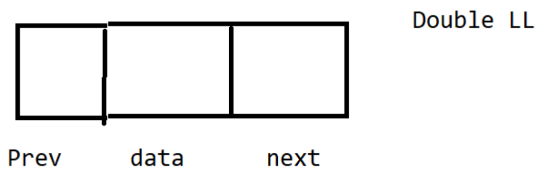
Linked List – Collection of Nodes. It may Single LL or Double LL. You can make both as Circular way. In that case last points the first.

DLL : Always first previous is null, and last node next is null.



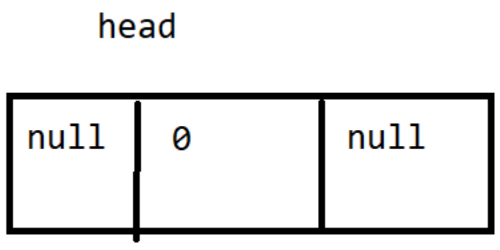


Understand the basics.

```
public class Node {
	Node prev;
	int data;
	Node next;

	@Override
	public String toString() {
		return data + " ";
	}

}

public class TestNode {
	public static void main(String[] args) {
		// I need a create header, but no data assigned to it.
		Node head = new Node();
		// Create a First Node names as n1
		Node n1 = new Node();
		n1.data = 10;
		n1.prev = head; // assign the reference of head to the n1.prev
		n1.next = null; // not needed, by default it assign null
		head.next = n1;
//		System.out.println(n1.prev);
		// Create a Node n2
		Node n2 = new Node();
		n2.data = 20;
		n2.prev = n1;
		n2.next = null;
		n1.next = n2;
//		System.out.println(n2.prev);
		//System.out.println(head.next.next.prev); //0  10  20
```

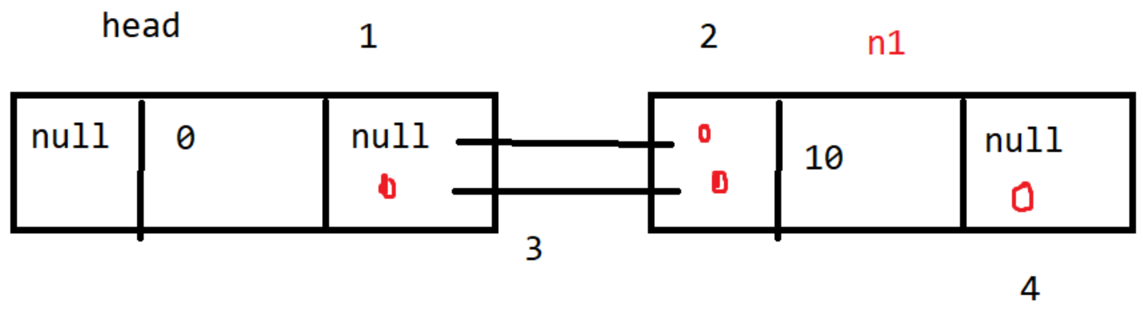
```
    // Add a node between n1 and n2
        Node mid = new Node();
        mid.data = 15;
        mid.prev = n1;
        mid.next = n2;
        n1.next = mid;
        n2.prev = mid;
        // How to process the list from the head.

        // Regular iteration

        int sum = 0;
        for(Node s = head.next; s!=null; s=s.next) {
            System.out.println(s);
            sum = sum+s.data;
        }
        System.out.println("Sum = " + sum);
        sum = 0;
        Node start = head.next;
        while(start!=null) {
            System.out.println(start);
            sum = sum+start.data;
            start = start.next;

        }
        System.out.println("Sum = " + sum);
    }
}
```

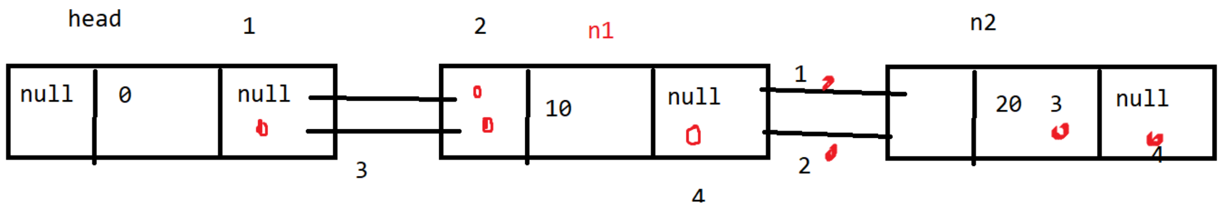
Refer: MyStringDLinkedList from DemoCode\lesson8-lists\list



No data assigned to the header. Always header is a reference to start.

1,2,3 and 4 are the links must connect. Need to write a Java code to make a connection between nodes using the node name.





```
class MyLinkedList{

Node header;

//Declare the Node structure as Inner class








}
```

Double Linked List User Implementation

- If you want to work with Linked list, you should only header reference.
- Through header only, you can be able to access your list item.
- Header does not hold the data. It is the starting point for your list.

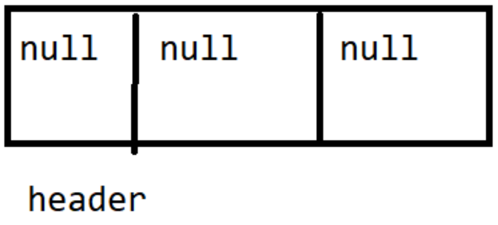
- We are doing position/index-based implementation.
- Position starts from zero.Always zero index item is the header.next(first item) if exists.
- Your Node structure has previous, value(data) and next.

1. Once you call the constructor it will initialize the header with null values of it's previous, value and next.

addFirst(String item)

- If you want to add in the front that will be first node (index zero)
- First node previous is always header.
- If header.next is null means no item in the list, add the new Node in the front with previous as header and next as null with the given value.
- If header.next is not null means, there are some items in the list. In that case add the new node with previous as header and next as header.next.previous as new node with its value.

```
MyStringDLinkedList ob = new MyStringDLinkedList();
```



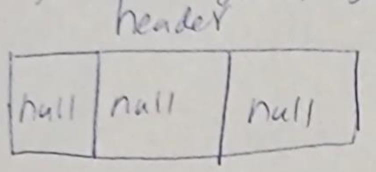
MyStringDLinked.java

addFirst (String item)

Possibility 1 : List is empty. [header.next == null]

Possibility 2 : List may contain one or more elements.

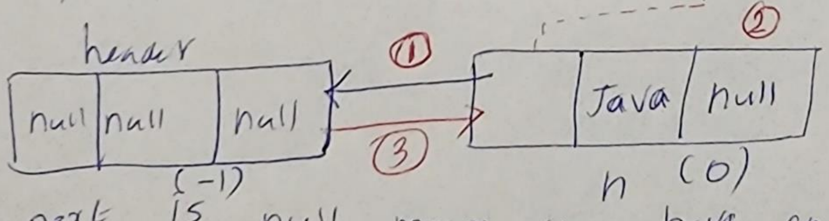
→ Input item passed as a String. Need to make a node for the given string.



Always if add a Node in the First position, The new Node previous always header and next is always header. next. So make a new Node with the given code. "Java"

Node n = new Node (header, item, header.next);

Picture representation of Possibility One .

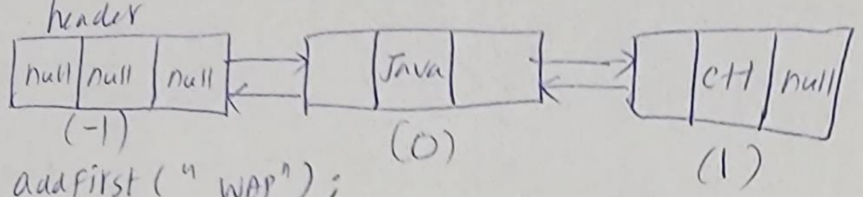


header.next is null means, you have an empty list. ① header ② header.next ③ header.next = n

① & ② connections are done through Node n creation.

Possibility 2   header. next != null

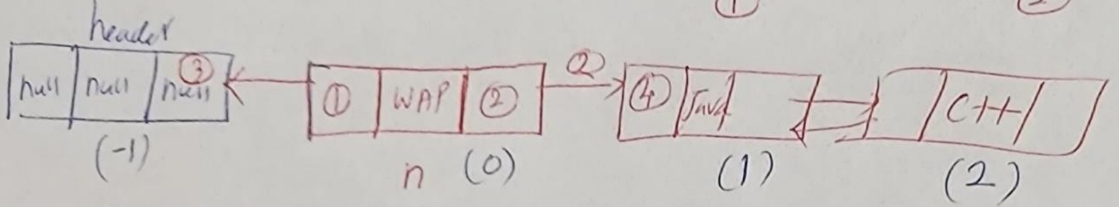
picture representation for possibility - 2



addFirst ("WAP");

Create a Node with the data "WAP"

Node n = new Node (header, item, header.next);
                               ①   ②



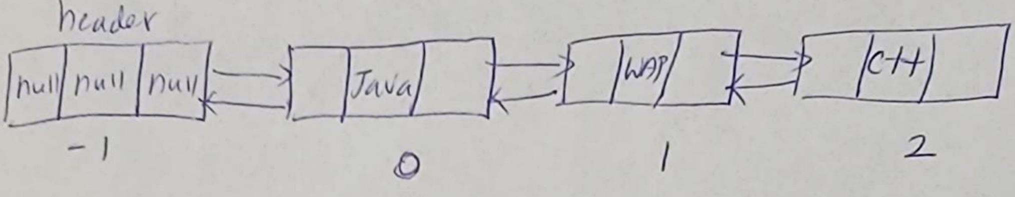
Connection ③ Code   header. next = n;

Connection ④ Code   header. next. previous = n;

insert(string s, int pos)

-> Traverse from header to position point.

picture represention with the below Scenario.



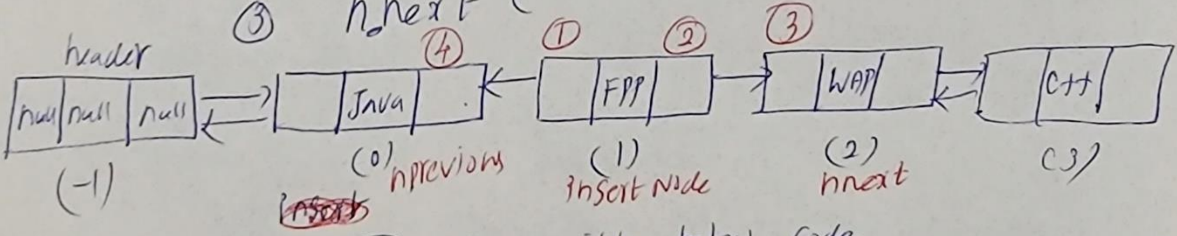
insert ("FPP", 1)

-> Identify the previous and next for the new Node to insert. For the above Scenario the new Node previous is pos (0) and next is pos (1). You have three named references

① New Node InsertNode

② nprevious

③ nnext (Make sure nnext is null or not)



Connection ① & ② done with below Code

```
Node insertNode = new Node ( nprevious, s, nnext );
```

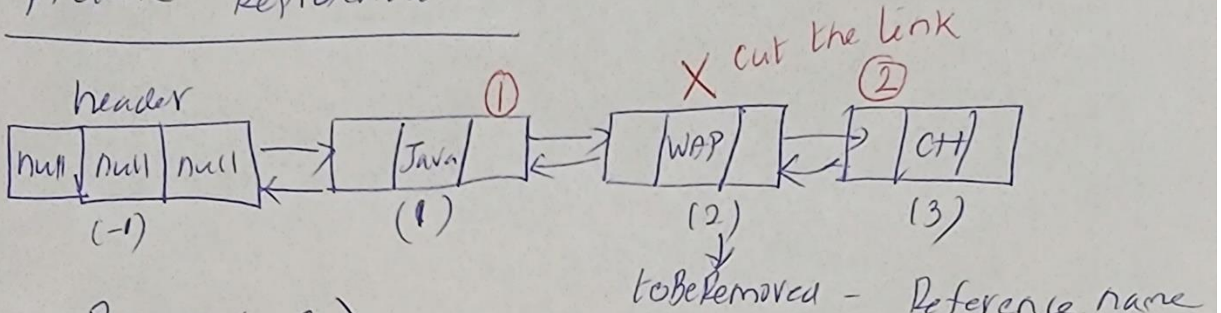
Connection ③ ~~n~~ next. previous = insertNode;

Connection ④ nprevious. next = insertNode;

boolean remove(int index)

→ Get the Node at index.
→ After retrieving Check the retrieved Node is null or not.

Picture Representation



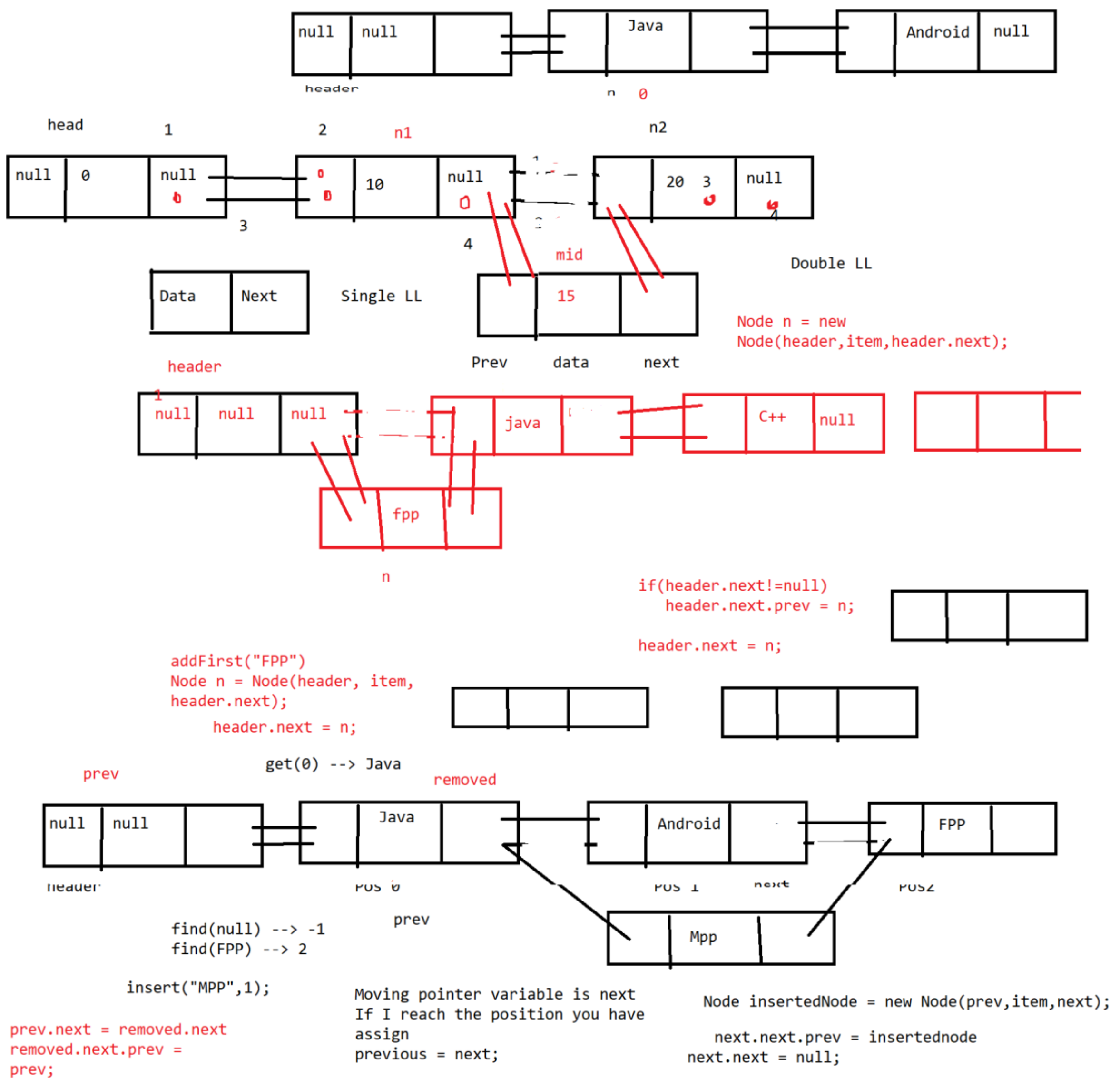
Remove (2)

→ To Remove Node (2), ~~Make~~ ~~to~~ Change the Connection of ① and ②

→ Make Sure toBoRemoved next is null or not.

Connection ① Code :

```
toBeRemoved. previous. next = toBeRemoved. next;
toBeRemoved. next. previous = toBeRemoved. previous;
```



```
Node n = new
Node(header,item,header.next);
```

```
if(header.next!=null)
   header.next.prev = n;

header.next = n;
```

```
addFirst("FPP")
Node n = Node(header, item,
header.next);
     header.next = n;
```

```
get(0) --> Java
```

```
find(null) --> -1
find(FPP) --> 2
```

```
insert("MPP",1);
```

```
prev.next = removed.next
removed.next.prev =
prev;
```

```
Moving pointer variable is next
If I reach the position you have
assign
previous = next;
```

```
Node insertedNode = new Node(prev,item,next);

  next.next.prev = insertednode
next.next = null;
```

To loop through your Java Collections you can use the following approaches

1. Index based for loop
2. For each
3. While

4. Iterator

Iterable and Iterarors are the Interface.

Any Collection implements Iterable you can use with for each

addFirst(String item)

When you try to add item in the front, either list is empty or may one more item.