

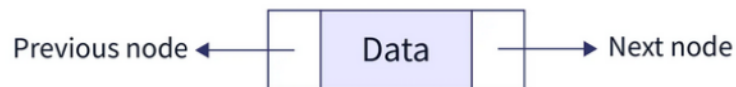
## Summary

### Sessions (09-02-2023)

- If you want to create a singly linked list using STL then they provide a `forward_list` library. You have just import it and you can use it using methods.

```
#include <iostream>
#include <forward_list>
using namespace std;
main () {
    forward_list<int> mylist = { 10, 20, 30, 40 };
    for (int temp: mylist) {
        cout << temp << endl;
    }
}
```

- As we already know in the singly linked list we can travel in only one direction which is forward direction. We can't go back if we use a singly linked list.
- If you want to travel both forward and backward direction then Doubly linked list comes into the picture.
- Doubly linked list consists of nodes which have data, a pointer to the next node, but also a pointer to the previous node.



- For creating doubly linked lists first we need to create a class for node.

```
class Node {  
    public:  
        Node *prev;  
        int data;  
        Node *next;  
};
```

- Create a first node :

```
Node *head = new Node;  
head -> data = NULL;  
head -> prev = NULL;  
head -> next = NULL;  
  
Node *A = new Node;  
A -> data = 10;  
A -> prev = NULL;  
A -> next = NULL;  
  
head -> next = A;  
A -> prev = head;
```

- Create a second node.

```
Node *B = new Node;  
B -> data = 20;  
B -> prev = NULL;  
B -> next = NULL;  
  
A -> next = B;  
B -> prev = A;
```