**Design Doubly Linked List - Solution**

[Report Issue](https://github.com/LeetCode-Feedback/LeetCode-Feedback/issues)

Let's briefly review the structure definition of a node in the doubly linked list.

A white rectangular object with black lines

AI-generated content may be incorrect.

Based on this definition, we are going to give you the solution step by step. The solution for the doubly linked list is similar to the one using singly linked list.

**1. Initiate a new linked list: represent a linked list using the head node.**

A white background with blue text

AI-generated content may be incorrect.

**2. Traverse the linked list to get element by index.**

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

**3. Add a new node.**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

Similar to the singly linked list, it takes O(N) time to get a node by index, where N is the length of the linked list. It is different from adding a new node after a given node.

**4. Delete a node.**

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Similar to the add operation, it takes O(N) time to get the node by the index which is different from deleting a given node. However, it is different to the singly linked list. When we get the node we want to delete, we don't need to traverse to get its previous node but using the "prev" field instead.