**Add Operation - Singly Linked List**

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

If we want to add a new value after a given node prev, we should:

1. Initialize a new node cur with the given value;A diagram of a diagram

   AI-generated content may be incorrect.
2. Link the "next" field of cur to prev's next node next;A diagram of a flowchart

   AI-generated content may be incorrect.
3. Link the "next" field in prev to cur.A diagram of a diagram

   AI-generated content may be incorrect.

Unlike an array, we don’t need to move all elements past the inserted element. Therefore, you can insert a new node into a linked list in O(1) time complexity if you have a reference to prev, which is very efficient.

*An Example*

A blue line with a square and a dot

AI-generated content may be incorrect.

Let's insert a new value 9 after the second node 6.

We will first initialize a new node with value 9. Then link node 9 to node 15. Finally, link node 6 to node 9.

After insertion, our linked list will look like this:

A diagram of numbers and arrows

AI-generated content may be incorrect.

*Add a Node at the Beginning*

As we know, we use the head node head to represent the whole list.

So it is essential to update head when adding a new node at the beginning of the list.

1. Initialize a new node cur;
2. Link the new node to our original head node head.
3. Assign cur to head.

For example, let's add a new node 9 at the beginning of the list.

1. We initialize a new node 9 and link node 9 to current head node 23.A diagram of a number

   AI-generated content may be incorrect.
2. Assign node 9 to be our new head.A blue arrow with black text

   AI-generated content may be incorrect.

What about adding a new node at the end of the list? Can we still use similar strategy?