Doubly-Linked List – allow us to move forward and backward through the list, all by by simply adding one extra pointer to our struct definition.

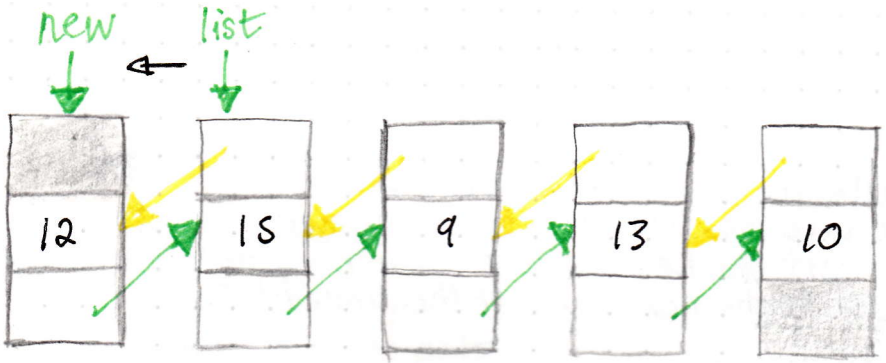
```
typedef struct dllist
{
    VALUE val;
    struct dllist *prev;
    struct dllist *next;
}
dllnode;
```

- Insert a new node into the linked list.

~~We go~~
~~steps~~ `dllnode **insert(dllnode* head, VALUE va);`

`list = insert(list, 12);`

Steps involved:
just like we did in singly-linked lists – expect here;
a. Fix the prev pointer of the old head of the linked list.



- Delete a node from a linked list.

`void delete(dllnode* target);`

`delete(x);`

Steps involved;
a. Fix the pointers of the surrounding nodes to "skip over" target.
b. Free target.