Singly vs. doubly linked lists

Selecting transcript lines in this section will navigate to timestamp in the video

- So far, we've been dealing with linked lists that have nodes with a piece of data and a next pointer. We call these singly linked lists, because they only have a pointer pointing to the next node in the list. With only a next pointer, you can only traverse forward through a list. You cannot go backwards. However, we can fix this by adding a previous pointer to each of our nodes. Now, each node of the linked list will contain a piece of data, a next pointer, as well as a previous pointer. We call this a doubly linked list, because we have a next and previous pointer and we can go through the list forward or backward. In Java, if we implement a doubly linked list from scratch, we just need to add a previous pointer to our node blueprint and a tail node to our linked list as a whole. By modifying each node within the linked list to have a previous pointer or previous node, we can change a linked list from singly to doubly linked.