1. A singly linked list is a type of linked list that consists of nodes containing data fields and a reference to the next node. The next-to-last of the last node would be null, making it the end of the list. You can also insert and delete nodes. Each node has a pointer or a tail with the function “Next” until it reaches “Null.” Meaning, singly linked list only has the Next function, and no way to go back.
2. We use singly linked lists when we need to make frequent insertions/deletions, when your data size is always changing since singly linked lists are dynamic, and when you need to implement queues or stacks of data, like how the music player works in Spotify, where what you click will play and the rest will wait in line. Then you use arrays when you need to quickly access an element using its index, when your data size is fixed or predictable, and when you want easier use and access. In short, singly linked lists are for when you are working on a complex and changing data set, and arrays are for when you need a quick and fast iteration of data, sometimes used on small projects.
3. Linked lists are used in browsers where we click the next page and previous page. It is also used in image viewers, where you can go to the next image and go back to the last. It can also be used for music players like in Spotify where we can move to the next music and back.

Reference:

- https://www.geeksforgeeks.org/dsa/singly-linked-list-tutorial/
- https://www.geeksforgeeks.org/dsa/linked-list-vs-array/
- https://www.naukri.com/code360/library/application-of-linked-list-data-structure