🌧️

ArrayList & LinkedList (Cheat Sheet)

Java Cheat Sheet: ArrayList & LinkedList Methods

📌 Use this as a quick reference while coding!

1️⃣ ArrayList Methods 🚀

(ArrayList is a **dynamic array**, good for fast random access)

Method	Description	Example
`add(E e)`	Adds an element to the end	`list.add("Apple");`
`add(int index, E e)`	Adds an element at a specific index	`list.add(1, "Banana");`
`get(int index)`	Retrieves an element at index	`list.get(2);`
`set(int index, E e)`	Replaces element at index	`list.set(0, "Mango");`
`remove(int index)`	Removes element at index	`list.remove(1);`
`remove(Object o)`	Removes first occurrence of object	`list.remove("Apple");`
`size()`	Returns number of elements	`list.size();`
`contains(Object o)`	Checks if list has an element	`list.contains("Mango");`
`indexOf(Object o)`	Returns index of first occurrence	`list.indexOf("Apple");`
`isEmpty()`	Checks if list is empty	`list.isEmpty();`
`clear()`	Removes all elements	`list.clear();`
`sort(Comparator c)`	Sorts elements	`Collections.sort(list);`
`reverse()`	Reverses the list	`Collections.reverse(list);`

2️⃣ LinkedList Methods 🔗

(LinkedList is **better for frequent insertions/deletions**)

Method	Description	Example

`add(E e)`	Adds element at end	`list.add("John");`
`addFirst(E e)`	Adds element at start	`list.addFirst("Alice");`
`addLast(E e)`	Adds element at end	`list.addLast("Bob");`
`get(int index)`	Retrieves element at index	`list.get(2);`
`getFirst()`	Retrieves first element	`list.getFirst();`
`getLast()`	Retrieves last element	`list.getLast();`
`remove(int index)`	Removes element at index	`list.remove(1);`
`removeFirst()`	Removes first element	`list.removeFirst();`
`removeLast()`	Removes last element	`list.removeLast();`
`size()`	Returns number of elements	`list.size();`
`contains(Object o)`	Checks if element exists	`list.contains("Alice");`
`isEmpty()`	Checks if list is empty	`list.isEmpty();`
`clear()`	Removes all elements	`list.clear();`

🔷 When to Use What?

✅ Use `ArrayList` when:

- You need **fast access** (`get()`, `set()`).
- The number of elements is **mostly fixed**.

✅ Use `LinkedList` when:

- You have **frequent insertions/deletions** (`addFirst()`, `removeFirst()`).
- You need a **queue or stack-like behavior**.

💡 **Tip:** If you're unsure, **start with** **`ArrayList`** and switch to `LinkedList` only if performance issues arise! 🚀🔥