

ArrayList vs LinkedList – Differences

Feature	ArrayList	LinkedList
Underlying Structure	Dynamic array	Doubly linked list
Access Time (get)	Fast ($O(1)$) – direct index access	Slow ($O(n)$) – must traverse nodes
Insert/Delete at Middle	Slow ($O(n)$) – elements shift	Fast ($O(1)$) – adjust links
Insert/Delete at End	Fast, except on resize	Fast
Memory Usage	Less memory	More memory (stores pointers)
Overall Performance	Better for searching and access	Better for frequent insert/delete

When to Use Which?

Use ArrayList when:

- You need **fast access** using index.
- Most operations are **read / get / iterate**.
- You rarely insert or delete elements in the **middle**.

Example use cases:

- Storing student records for reading
 - Maintaining a list of items to display on UI
 - Working with data where random access is required
-

Use LinkedList when:

- You need **frequent insertion/deletion**, especially at the **beginning or middle**.
- You are implementing **queues or dequeues**.

Example use cases:

- Music playlist where songs are added/removed often
- Browser history (back/forward list)
- Implementing Queue, Deque