

# ArrayList vs LinkedList – Differences

Feature	ArrayList	LinkedList
<b>Underlying Structure</b>	Dynamic array	Doubly linked list
<b>Access Time (get)</b>	<b>Fast (<math>O(1)</math>)</b> – direct index access	<b>Slow (<math>O(n)</math>)</b> – must traverse nodes
<b>Insert/Delete at Middle</b>	<b>Slow (<math>O(n)</math>)</b> – elements shift	<b>Fast (<math>O(1)</math>)</b> – adjust links
<b>Insert/Delete at End</b>	Fast, except on resize	Fast
<b>Memory Usage</b>	Less memory	More memory (stores pointers)
<b>Overall Performance</b>	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 deques**.

**Example use cases:**

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