Great topic bro! 🔥 Let's break it down clearly and practically:

**🔺 What is Thread.interrupt()?**

interrupt() is a way to **politely** ask a thread to stop what it’s doing.

But it **does NOT forcefully kill** the thread.

**✅ When you call thread.interrupt():**

You’re setting an **"interrupted flag"** on the thread (a boolean inside thread metadata).

**🧠 What happens next depends on the thread's state:**

**🟢 1. If the thread is RUNNING (normal work)**

Then nothing immediately happens. You have to **manually check** if it's interrupted:

while (!Thread.currentThread().isInterrupted()) {

// Keep working

}

System.out.println("Stopped gracefully");

**🟡 2. If the thread is BLOCKED (sleep, wait, join)**

Then it **immediately throws InterruptedException** and exits the blocked state.

Example:

Thread t = new Thread(() -> {

try {

Thread.sleep(5000); // Blocking state

} catch (InterruptedException e) {

System.out.println("Thread was interrupted during sleep!");

}

});

t.start();

Thread.sleep(1000);

t.interrupt(); // Interrupts during sleep

📌 So in blocking methods, **the only way to escape** is by getting interrupted → leads to InterruptedException.

**🔁 Thread Lifecycle + Interrupt:**

| **Thread State** | **Effect of interrupt()** |
| --- | --- |
| **New** | No effect |
| **Runnable** | Sets the interrupt flag ✅ |
| **Blocked** | If in sleep() / join() / wait() → throws InterruptedException |
| **Waiting** | Same: InterruptedException |
| **Terminated** | No effect |

**💡 How to check interruption manually?**

Thread.currentThread().isInterrupted(); // true or false

**❗Important Interview Notes:**

* interrupt() doesn’t **stop** the thread — it’s a **signal**.
* Thread must cooperate by checking isInterrupted() or catching InterruptedException.
* Often used to stop **long-running loops** or **background tasks** gracefully.
* Used in real-world: thread pools, async tasks, cleanup handlers, etc.

Want me to show you a full practical example of how to interrupt a long-running task gracefully?