Great question! Let's make this **crystal clear** by comparing **Q-learning** and **SARSA** with a **real-life analogy**, and then tie it back to how they work in RL.

**🤔 Real-Life Analogy: Navigating a Maze in the Dark**

Imagine you're blindfolded and navigating a maze for the first time. You're learning how to reach the end safely — and you're learning from **experience**.

**🧠 Q-Learning: “Learn from the *best possible* move”**

You *imagine* the best move you **could** have made from your next position — even if you don’t take it.

**🟩 Example:**

You're at a T-junction. You decide to go **left** and bump into a wall (bad choice).  
Q-learning says:

“Yeah, that was a bad move, **but** if I had gone **right**, it would’ve been amazing — let me learn from that imagined path instead!”

✅ **Learns the optimal path** quickly  
❌ Can overestimate how good a move is because it assumes you’ll *always* make the best move next time (even if you won’t).

**🧠 SARSA: “Learn from what you actually did”**

You **only learn** from the path you *actually took*, not what you *could’ve* done.

**🟨 Example:**

You're at that same T-junction. You again go **left** and hit the wall.  
SARSA says:

“Okay, that was bad, and I really did it. Let me adjust my learning based on **what I actually did**, not on some perfect world where I turned right.”

✅ More **realistic and safer learning**  
❌ Might take longer to find the *best* path because it learns cautiously

**🔁 TL;DR Comparison Table**

| **Feature** | **Q-Learning** | **SARSA** |
| --- | --- | --- |
| Type | Off-policy (learns from best *possible* action) | On-policy (learns from *actual* action taken) |
| Behavior | Optimistic | Realistic/Cautious |
| Analogy Summary | “What if I had gone the best way?” | “What actually happened when I moved?” |
| Learning Style | Thinks ahead assuming perfect play | Learns from its own real experience |
| Risk Handling | Can be riskier | Safer/adaptive |

**🤖 In GridWorld Terms:**

You're training a robot in a maze:

* **Q-Learning**: If the robot moves wrong, it still updates the table assuming the **best move** was taken next.
* **SARSA**: If the robot moves wrong, it only learns based on the **actual move** it did next — whether it was good or not.

Would you like me to write up **SARSA Level 1 code** now for your GridWorld setup, just like Q-learning?