

This DSL is a minimal, event-driven language tailored for IoT workflows. It allows users to declare event listeners using **on**, define conditions and actions as **lambda ... end** blocks, and react to events triggered with the syntax **"event" -> value**. The DSL operates around named events, single-parameter conditions, and simple actions, enabling users to express behaviors like turning on a fan when the temperature exceeds a threshold or activating an alarm when light levels drop.

Control flow is handled explicitly with **if ... end** for conditional execution and **for ... end** for loops. Input can be simulated or collected interactively using **input()**, and commands can be sent to devices via **send_command("device", "command")**. **lambda** blocks receive their arguments directly — such as *temperature* or *humidity* — without needing to work on dictionaries, which keeps the syntax clean and intuitive.

This DSL is not tied to a specific hardware platform but is designed to express common IoT patterns declaratively. It combines clarity, minimal syntax, and sufficient expressiveness to define rules for sensors, actuators, and workflows. Its use of explicit **end** delimiters for blocks, direct event triggering with **"event" -> value**, and IoT-specific commands like **send_command** makes it approachable for developers and domain experts building simple IoT automation scripts.