What Is an AI Agent?

What defines an AI agent: the observe-decide-act loop, how it differs from a single tool call, and the autonomy spectrum.

Intermediate 18 min

Last updated: August 22, 2026
Read online: https://www.learnforgex.com/en/topics/what-is-an-ai-agent

---

What Is an AI Agent?

This category has been referenced from several places already, without ever being defined. "Tools and Function Calling" drew a line between a single tool call and what it called, without defining, "a broader system that can plan multiple steps toward a goal" (see its "Tool Use vs. Agents" section). "Three Types of Learning: Supervised, Unsupervised, Reinforcement" in "Machine Learning" noted that reinforcement learning's *agent* is conceptually the closest existing idea to this one. "What Generative AI Is Not" pointed out that generating a paragraph or an image on a single prompt is not, by itself, "agentic." This lesson finally answers the question all three were pointing at: what specifically makes a system an AI agent, as opposed to a model, a chatbot, or a single tool call?

What Is an AI Agent?

An **AI agent** is a system -- built around one or more calls to a model, plus tool use -- that pursues a goal by repeatedly deciding what to do next, taking an action, observing the result, and deciding again, continuing until the goal is met or it decides to stop. Three things distinguish this from what "Tools and Function Calling" already covered: the *sequence* of decisions is not fixed in advance by a human, the *number* of steps isn't known ahead of time, and each decision is informed by what happened in the previous step rather than a single, one-shot prompt. A single tool call answering one question is not an agent by this definition; a system that decides, on its own, to look something up, check whether the result answers the question, and look up something else if it doesn't, is.

Why Does It Exist?

Some problems simply cannot be solved with one prompt and one tool call, because the right sequence of steps isn't known until earlier steps have already run. Debugging a failing test means trying something, observing whether it worked, and deciding what to try next

based on that outcome. Researching a topic across multiple sources means reading one, deciding whether it answered the question, and searching further if it didn't. Completing a multi-step task -- book a flight only if it's under a given price, otherwise check a different date -- requires a decision that depends on a result that doesn't exist yet when the task starts. None of these fit the tool-calling loop described in "Tools and Function Calling", where one request goes out, one result comes back, and a human is still driving each next step. Agents exist to handle exactly this class of problem: goals where the necessary steps, and how many of them, can only be determined along the way.

The Agent Loop: Observe, Decide, Act

An agent's execution is a loop, not a single request/response exchange:

1. **Observe** -- the agent looks at its goal and everything that has happened so far in this run: prior decisions, tool calls, and their results.
2. **Decide** -- based on that observation, the agent (through a model call) decides on exactly one next step: call a specific tool with specific arguments, or conclude that the goal is satisfied and produce a final answer.
3. **Act** -- if a tool was chosen, it's actually executed (by the surrounding application, exactly as "The Tool-Calling Loop" described -- the model itself still never runs anything), and its real result becomes part of the next observation.

This is a direct generalization of "The Tool-Calling Loop": instead of running once and stopping, the same observe-decide-act cycle repeats, with each iteration informed by everything that came before it, until the "Decide" step concludes the goal is met (or, as "Controlling Agent Behavior" later in this category covers, until a safety limit is reached).

From a Single Tool Call to an Agent

"Tool Use vs. Agents" already drew this line at a high level; it's worth restating precisely now that the loop above is on the table. Using a tool once, inside a single request/response exchange, is not what makes something an agent -- an ordinary chatbot that looks up today's weather when asked does not become an agent by doing so. What makes a system an agent is that the loop from the previous section runs under the *system's own control*: it, not a human, decides which tool to call, when to call another one, and when enough information has been gathered to stop -- sustained across multiple steps, toward a goal that may take an unpredictable number of steps to reach.

The Autonomy Spectrum

"Agent" doesn't describe one fixed amount of independence -- it names a spectrum. At one end, a system's every decision is effectively scripted by a human in advance (a single tool call is close to this end). At the other, a system runs many decide-act cycles entirely on its own, checking in with no one until it's done. Most agents that are actually useful in practice

sit somewhere between these extremes: some decisions are left fully to the loop above, while others -- anything costly, irreversible, or risky -- are deliberately routed back to a human before proceeding. Exactly where a given agent should sit on this spectrum, and how to enforce that choice in code, is the subject of "Controlling Agent Behavior" later in this category.

Where This Category Goes From Here

This lesson defined the loop; the rest of the category builds on it. "Agent Planning and Reasoning Patterns" goes deeper into the "Decide" step above -- concrete patterns (like ReAct, plan-and-execute, and reflection) for how that decision actually gets made, beyond "the model decides." "Controlling Agent Behavior" covers the guardrails a loop like this needs before it's safe to run with any real autonomy: step limits, human approval points, and observability. "Building an AI Agent in TypeScript" then builds a real, running agent loop -- reusing the exact MCP tools `GeoFactsServer.ts` exposed in "Building an MCP Server" -- to make everything covered here concrete.

Best Practices

- Reserve the word "agent" for a system that actually runs the loop from "The Agent Loop: Observe, Decide, Act" under its own control -- see "From a Single Tool Call to an Agent" for why a single tool call doesn't qualify.
- Before building an agent-shaped feature, place it on "The Autonomy Spectrum" deliberately -- deciding how much independence it actually needs shapes the rest of its design.
- Design around the loop, not a single pass through it -- a step's result is expected to change what happens next, not just get appended to a transcript.

Common Mistakes

- **Calling any system that uses a tool an "agent."** As "Tool Use vs. Agents" and "From a Single Tool Call to an Agent" both explained, a single tool call inside an otherwise ordinary exchange isn't an agent -- the defining feature is the self-directed loop, not tool use by itself.
- **Assuming more autonomy is automatically better.** "The Autonomy Spectrum" is a spectrum for a reason -- more unsupervised steps mean more chances for a costly or hard-to-reverse mistake, which is exactly what "Controlling Agent Behavior" exists to manage.
- **Mistaking an agent's ability to plan steps for genuine reasoning or understanding.** The model choosing the next step is still the same kind of model covered in "Reasoning Limits" in "LLM Capabilities and Limitations" -- planning a sequence of tool calls doesn't remove those underlying limits.

Summary, Cheat Sheet, and Glossary

Summary

- An AI agent is a system that pursues a goal by repeatedly deciding what to do next, acting, and observing the result -- not a single tool call or a single model response.
- Agents exist for problems where the right sequence of steps (and how many are needed) can't be known in advance, and depends on the outcome of earlier steps.
- The agent loop -- observe, decide, act -- is a direct generalization of "The Tool-Calling Loop" from "Tools and Function Calling", repeated under the system's own control instead of running once.
- What separates an agent from a single tool call is that loop running autonomously, not the mere presence of a tool call.
- Autonomy is a spectrum, not a binary -- most practical agents mix self-directed steps with deliberate human checkpoints, the subject of "Controlling Agent Behavior".

Cheat Sheet

- Agent = goal + observe-decide-act loop, run under the system's own control, not a human, across multiple steps.
- Loop = generalized tool-calling loop: observe (goal + history) -> decide (model picks next action) -> act (real tool execution) -> repeat.
- Single tool call != agent. The loop running on its own, across steps, is what qualifies.
- Autonomy spectrum: fully scripted <-> fully autonomous; most real agents sit in between.

Glossary

- **AI agent:** a system that pursues a goal by repeatedly deciding on and taking actions, observing their results, and deciding again, with some degree of autonomy.
- **Agent loop:** the observe-decide-act cycle an agent repeats until its goal is met or it's stopped.
- **Autonomy spectrum:** the range from fully human-scripted behavior to fully unsupervised, multi-step behavior that an agent's design can sit anywhere along.