



# LangChain overview

[Copy page](#)

LangChain is an open source framework with a pre-built agent architecture and integrations for any model or tool — so you can build agents that adapt as fast as the ecosystem evolves

LangChain is the easiest way to start building agents and applications powered by LLMs. With under 10 lines of code, you can connect to OpenAI, Anthropic, Google, and [more](#). LangChain provides a pre-built agent architecture and model integrations to help you get started quickly and seamlessly incorporate LLMs into your agents and applications.

We recommend you use LangChain if you want to quickly build agents and autonomous applications. Use [LangGraph](#), our low-level agent orchestration framework and runtime, when you have more advanced needs that require a combination of deterministic and agentic workflows, heavy customization, and carefully controlled latency.

LangChain [agents](#) are built on top of LangGraph in order to provide durable execution, streaming, human-in-the-loop, persistence, and more. You do not need to know LangGraph for basic LangChain agent usage.

## Create an agent





## LangChain overview

```
const getWeather = tool(
  ({ city }) => `It's always sunny in ${city}!`,
  {
    name: "get_weather",
    description: "Get the weather for a given city",
    schema: z.object({
      city: z.string(),
    }),
  },
);

const agent = createAgent({
  model: "claude-sonnet-4-5-20250929",
  tools: [getWeather],
});

console.log(
  await agent.invoke({
    messages: [{ role: "user", content: "What's the weather in Tokyo?" }],
  })
);
```

See the [Installation instructions](#) and [Quickstart guide](#) to get started building your own agents and applications with LangChain.

## Core benefits





## LangChain overview

Different providers have unique APIs for interacting with models, including the format of responses. LangChain standardizes how you interact with models so that you can seamlessly swap providers and avoid lock-in.

[Learn more >](#)

LangChain's agent abstraction is designed to be easy to get started with, letting you build a simple agent in under 10 lines of code. But it also provides enough flexibility to allow you to do all the context engineering your heart desires.

[Learn more >](#)

### Built on top of LangGraph

LangChain's agents are built on top of LangGraph. This allows us to take advantage of LangGraph's durable execution, human-in-the-loop support, persistence, and more.

[Learn more >](#)

### Debug with LangSmith

Gain deep visibility into complex agent behavior with visualization tools that trace execution paths, capture state transitions, and provide detailed runtime metrics.

[Learn more >](#)

[Edit this page on GitHub](#) or [file an issue](#).



[Connect these docs](#) to Claude, VSCode, and more via MCP for real-time answers.





## LangChain overview



### Resources

[Forum](#)

[Changelog](#)

[LangChain Academy](#)

[Trust Center](#)

### Company

[About](#)

[Careers](#)

[Blog](#)

Powered by [mintlify](#)





☰ LangChain overview

---

