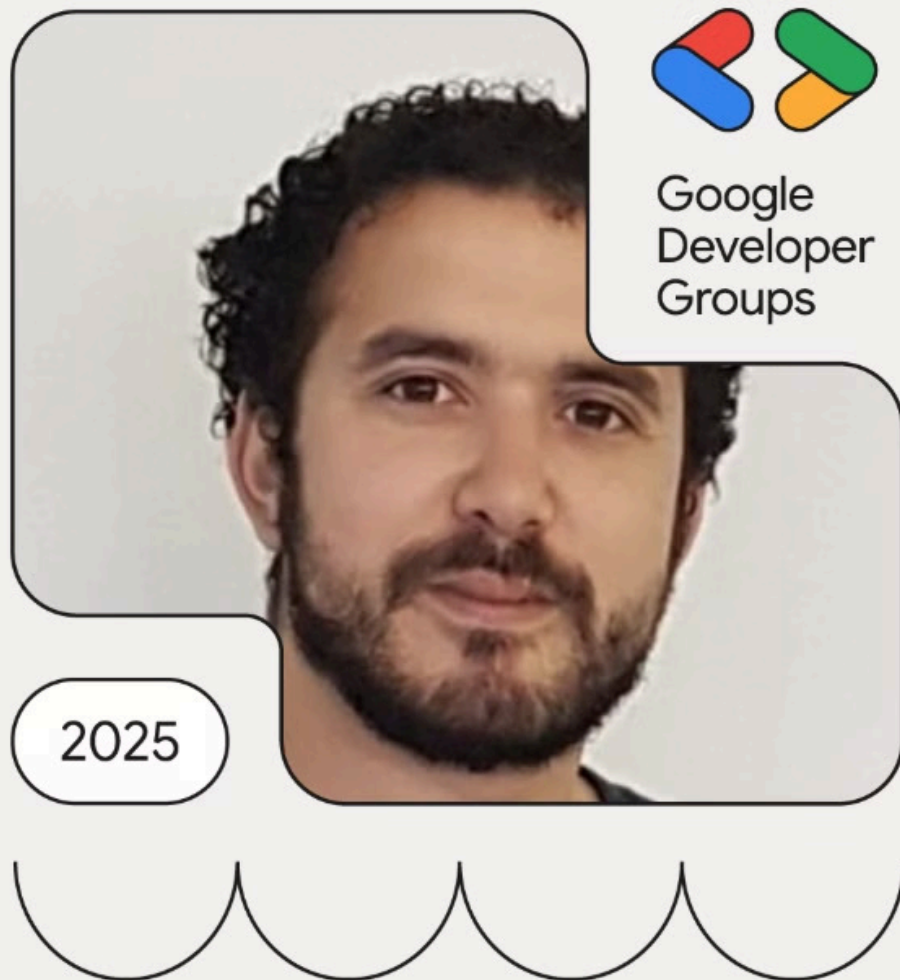




Workshop: Incremental
design of LLM-powered
agentic applications.

Yacine Touati

Software Developer / IT Consultant



Personal presentation



1

👋 Hi there!

I'm Yacine, a software developer and an enthusiastic futurist.

3

Background

Transitioned to software development in 2018 after a decade in the French Ground Army.

2

Passions

Tech, AI, software engineering, web development.

4

Specialization

Full-stack development for web (and now, agentic) applications.

Workshop Overview



Module 1

Basics of instrumenting LLMs with Gemini.



Module 2

Call tools with Gemini.



Module 3

The ReAct pattern.



Module 4

The Plan and Execute pattern.



Module 5

Multi agent systems.

Module 1: Instrumenting LLMs Basics with Gemini and LangChain

This module will introduce you to the fundamentals of instrumenting large language models (LLMs) with Gemini and LangChain. We'll cover essential concepts and practical techniques to enhance your understanding of these powerful technologies.





LangChain



python.langchain.com



Introduction | 🦜🔗 LangChain

LangChain is a framework for developing applications powered by large language models (LLMs).



langchain-ai.github.io

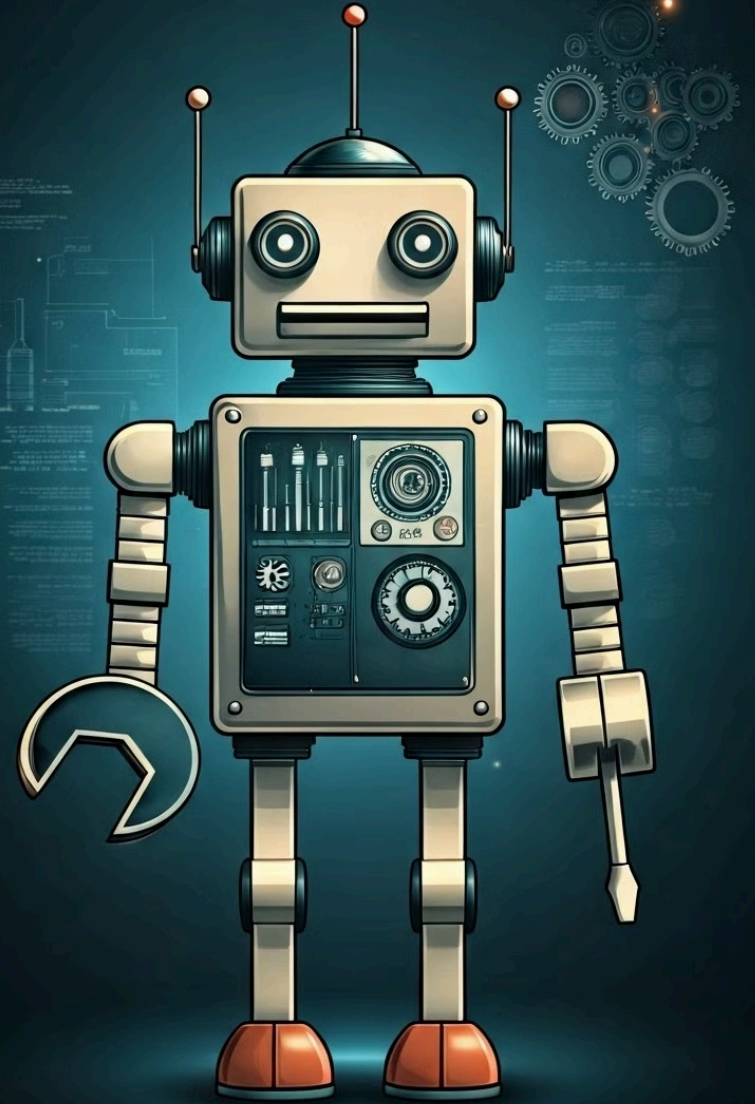


Home

Build language agents as graphs

Module 2: Call External Tools with LLMs

Extend LLM capabilities by connecting them to external APIs and resources, enabling them to interact with the real world.



Calling Tools: Extending Agent Capabilities

In agentic applications, "calling tools" is how agents access and utilize external capabilities to enhance their problem-solving and task execution. Think of "tools" as extensions that provide specialized functionalities beyond the agent's core programming.

Access Information

Search the web, browse specific websites, query databases.

Interact with the Real World

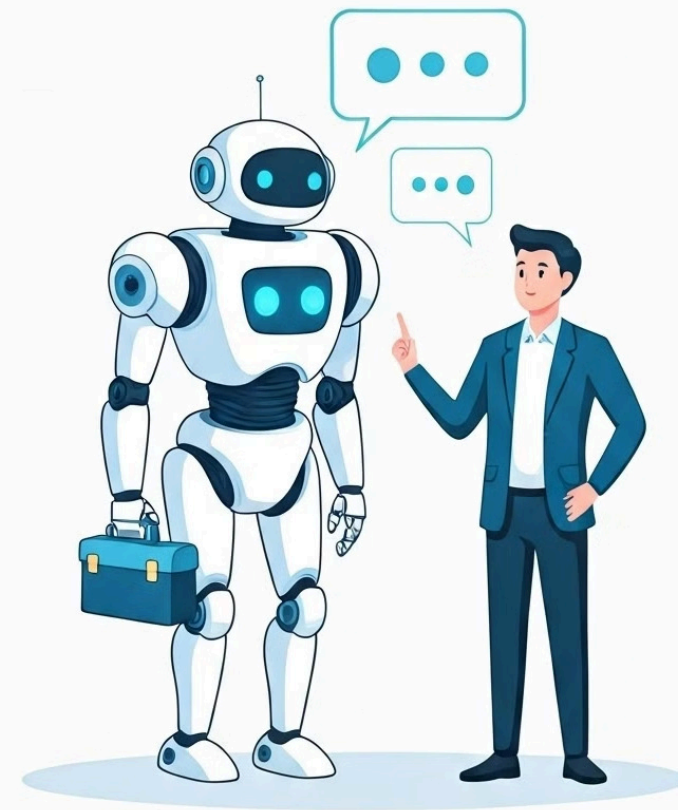
Control devices, send emails, schedule events.

Perform Complex Operations

Translate languages, perform calculations, generate images.

Why is this important? Tool use is key to building agentic applications that are adaptable and can interact with diverse environments and data sources.

Module 3: The ReAct Pattern



ReAct Pattern: Tool Use for Agentic Applications

The ReAct pattern allows agents to reason and act effectively, especially when using external tools to enhance their capabilities.

Analyze Input: Tool Needed?

Agent assesses if a tool is required to address user input effectively. Example: "Weather in Paris?" needs a tool.

Use Tool (if needed)

Agent selects and employs the appropriate tool (e.g., search, maps) to gather necessary information.

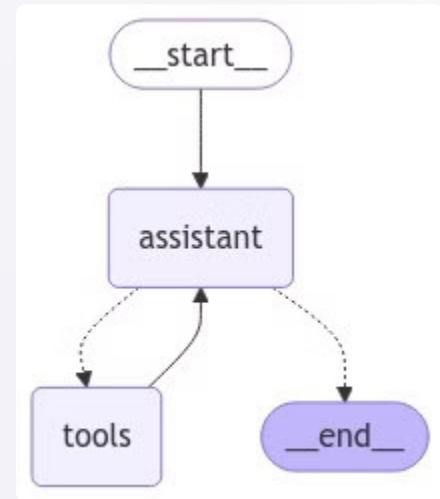
Answer with Tool Output

Agent integrates tool output with its model to formulate a comprehensive response.

Direct Answer (if no tool needed)

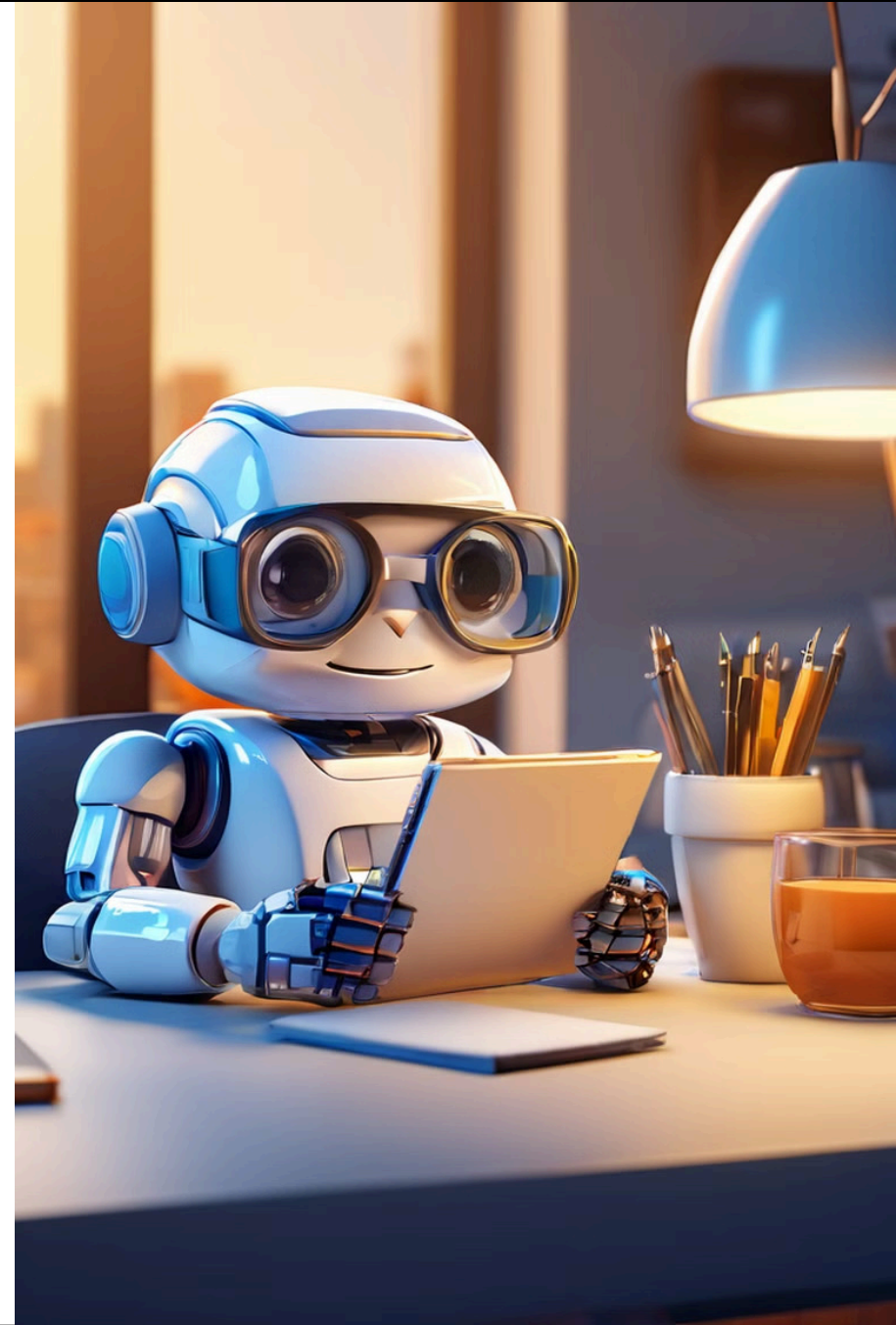
If no tool is necessary, the agent uses its model directly to answer the user's input.

ReAct enables agents to handle requests beyond simple questions, access real-world data, and perform complex tasks effectively, making them more powerful and adaptable.

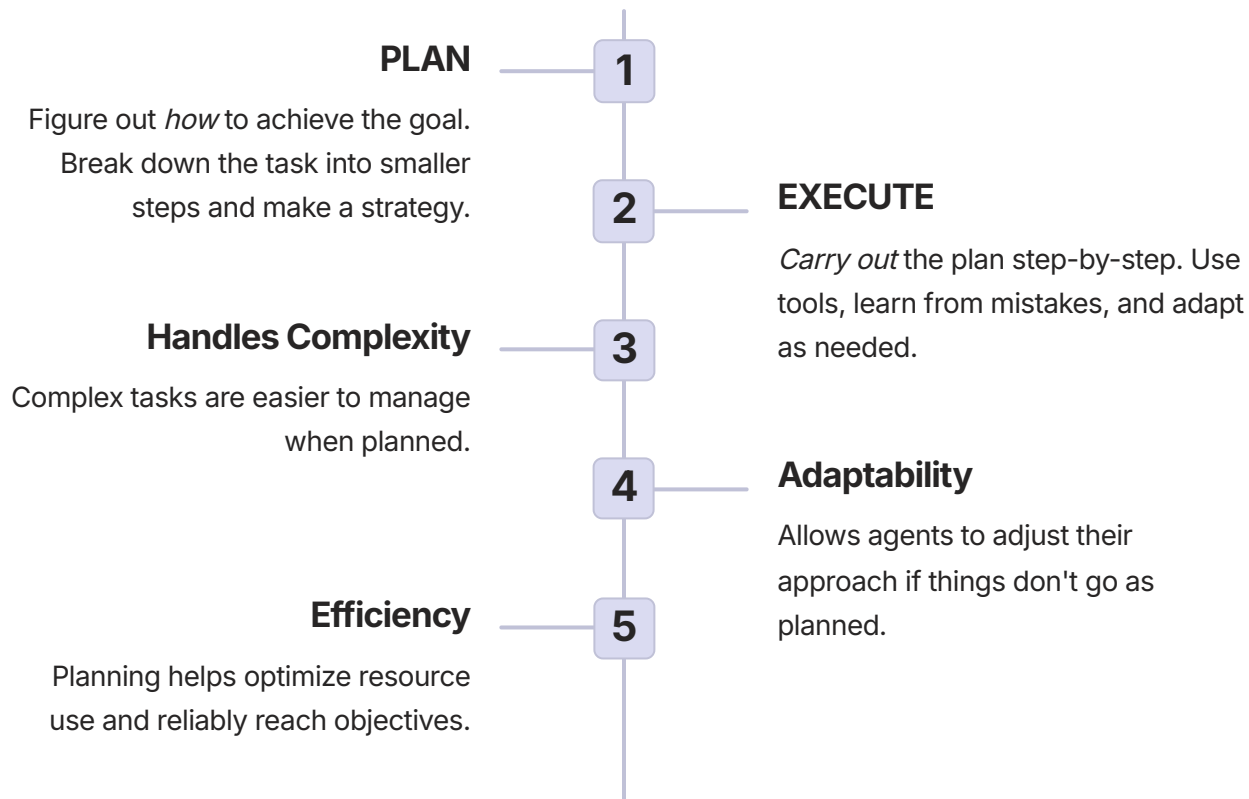


Module 4: The Plan and Execute Pattern

Extend the ReAct pattern for more complex tasks by incorporating planning and execution steps.



Plan and Execute Pattern in Agentic Applications



Takeaway: "Plan and Execute" empowers agentic applications to be smart, adaptable problem-solvers, not just simple instruction followers.



Module 5: Multi-Agent Systems

Explore the capabilities of multi-agent systems for enhanced collaboration and problem-solving in complex tasks.



Multi-Agent Collaboration

Solving complex problems through teamwork.

Instead of one agent doing everything, we have multiple agents, each with specific skills, working together.

The key idea here is that agents share a common scratchpad of messages, so that each agent know what the other agents are doing.

Collaboration via Shared Scratchpad examples:

- travel planning
- deep research
- content creation

Benefit: Each agent focuses on its strength, leading to a more robust and efficient application!

