

Practical Agentic AI: Tools and Python Development

Two days from guided tool use to a bounded agent prototype

FA-0557 · 2 days

Agentic & Generative AI · Software Development

Explore agent tools and research workflows, then build a small Python/LangChain agent with tool integration, context handling and review boundaries.

Introduction

This two-day course connects practical agent-tool use with introductory agent development. Day 1 is accessible to non-technical participants and covers selected task and research workflows, prompting and business use cases.

Day 2 requires Python and API proficiency. Participants examine an agent loop and build a small, bounded prototype using LangChain, selected tools and a compatible model. Advanced features are introduced through examples rather than a complete production deployment. Non-technical participants should not expect to complete the development labs without the stated prerequisites.

Tool demonstrations depend on account access and current capabilities. Exercises use illustrative, non-confidential data and reversible sandbox tasks; outputs and proposed external actions require review.

Learning outcomes

The course teaches participants to:

Day 1 — Practical use

- Explain tool-using agents, capabilities and limitations.
- Use selected ChatGPT Work/research and Manus workflows where available.

- Write task prompts with scope, constraints and review requirements.
- Identify suitable pilot use cases and evaluate results.

Day 2 – Development

- Explain the model/tool loop, context and selected planning patterns.
- Build a small LangChain/Python agent using compatible current interfaces.
- Connect an API or tool and handle selected errors and context requirements.
- Review a prototype for privacy, security, bias and human oversight.

Prerequisites

Day 1

- Basic computer literacy.
- Familiarity with web browsing and common software applications.

Day 2

- Proficiency in Python programming.
- Understanding of APIs and web development concepts.
- Experience with LLMs and AI frameworks is beneficial.

A laptop and access to the selected tools or model endpoints. Confirm versions, accounts and any usage costs before the course.

Training outline

1. Day 1 – Agentic AI for practical users

1. Introduction to Agentic AI

- Definition and evolution of agentic AI.
- Illustrative applications and case discussions.
- Overview of selected ChatGPT Work/research and Manus capabilities, access and limitations.

2. Tool-supported task execution

- Explore selected ChatGPT Work capabilities where available.
- Practise an illustrative form/task workflow in a sandbox; review proposed appointments or purchase actions without making real bookings or purchases.
- Set scope, permission boundaries and checks for tool outputs.

3. AI-assisted research

- Explore research workflows, including the Deep research plugin where available.
- Review sourced reports, inspect citations and check that evidence supports conclusions.
- Integrating Deep Research outputs into decision-making processes.

4. Manus AI

- Introduction to Manus task execution and its sandbox environment.
- Selected demonstrations in website creation, data analysis or content generation; review the resulting work.
- Comparative analysis with other AI agents.

5. Prompt Engineering Essentials

- Crafting effective prompts to guide AI behavior.
- Understanding the impact of prompt structure on outcomes.
- Interactive session: Refining prompts for desired results.

6. Practical Applications and Use Cases

- Identifying opportunities for agentic AI in various industries.
- Developing strategies to integrate AI agents into daily workflows.
- Group discussions: Sharing ideas and experiences.

2. Day 2 – Building a bounded agent with Python

1. Deep Dive into Agentic AI Architecture

- Understanding the components of AI agents: LLMs, tools, memory, and planning.
- Exploring the ReAct pattern for decision-making.

- Trace information, model decisions and tool actions through an agent loop.

2. Introduction to LangChain

- Set up a compatible current LangChain/Python environment.
- Build a simple model/tool workflow; compare a fixed chain with an agent loop.
- Use selected supported LangChain integrations and the current `create_agent` interface.

3. Developing Custom Agents

- Defining agent behavior and goals.
- Integrating external APIs and tools for enhanced capabilities.
- Implementing memory and context management.
- Error handling and fallback strategies.

4. Selected advanced features

- Incorporating real-time data retrieval and processing.
- Enabling multi-step reasoning and planning.
- Run a bounded agent prototype for a selected data-analysis, content or automation task; distinguish a lab from production deployment.

5. Ethical Considerations and Best Practices

- Identify potential bias and evaluate fairness issues in example outputs.
- Implementing security measures and data privacy protocols.
- Establishing guidelines for responsible AI agent deployment.

6. Hands-On Project

- Design and develop a small, scoped agent prototype for a selected illustrative use case.
- Presentations and peer reviews to provide feedback and insights.