How to Build AI Agents from Scratch - Even If You've Never Done It Before.

This is Maryam Miradii's 10-step roadmap from prompt to evaluation.

Step 1: Define the Agent's Role and Goal
+ What will your agent do?
* Who is it helping?
+ What kind of output will it generate?
- Example: A medical assistant agent that reads X-rays, summarises findings, and speaks results.

Step 2: Design Structured Input & Output
+ Use Pydantic AI or JSON Schemas to define what the agent receives and returns.
+ Avoid messy text — think like an API.
- Tools: Pydantic AI, LangChain Output Parsers

Step 3: Prompt, Tune, and Define Agent Protocol
+ Start with role-based system prompts.
+ Use Prompt Tuning or Prefix Tuning for consistent persona and task behaviour.
+ Use MCP to standardise how your agent handles inputs, tools, and outputs across modules.
- Tools: GPT-4, Claude, MCP, Prompt Tuning

Step 4: Add Reasoning and Tool Use
+ Equip the agent with reasoning frameworks:
> ReAct (Reasoning + Action)
> Chain-of-Thought
+ Allow access to tools like web search, code interpreters, or document retrievers.
- Tools: LangChain, OpenAI Tools, ReAct Framework

Step 5: Structure Multi-Agent Logic (if needed)
+ Use orchestration frameworks to define agent roles and coordination.
+ Create Planner, Researcher, Reporter agents - each with its own input/output schema.
- Tools: CrewAI, LangGraph, OpenAI Swarm

Step 6: Add Memory and Long-Term Context
+ Does your agent need to remember what happened earlier?
+ Use conversational memory, summary memory, or vector-based memory.
- Tools: Zep, LangChain Memory, Chroma

Step 7: Add Voice or Vision Capabilities (optional)
+ Text-to-speech: Use Coqui or ElevenLabs.
+ Image understanding: Use GPT-4 or LLaMA 3.2 Vision.
- Let your agent see and speak.

Step 8: Deliver the Output (in human or machine format)
+ Format outputs into Markdown - PDF or structured JSON.
+ Output must be both readable and parsable.

- Tools: Pydantic AI, Markdown-to-PDF, LangChain Output Parsers

Step 9: Wrap in a UI or API (optional)
+ Create a front-end or expose your agent via API.
+ Use Gradio, Streamlit, or FastAPI.
- This is what turns your agent into a product.

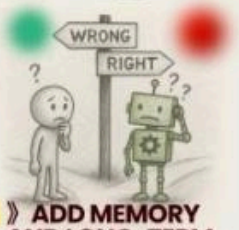
Step 10: Evaluate and Monitor Your Agent's Performance
+ Run test prompts and toolchains to ensure reliability and consistency.
+ Use logs, benchmarks, and feedback loops to improve over time.
- Tools: MCP Logs, OpenAI Evaluation API, Custom Metrics Dashboards

Credits to Maryam Miradi, PhD. Follow her for more. Linkedin, YouTube, X: @maryammiradi

HOW TO BUILD AI AGENTS from Scratch

Dr. Maryam Miradi

@maryammiradi



Step 1 》 DEFINE THE AGENT'S ROLE AND GOAL

* What will your agent do?
* Who is it helping?
* What kind of output will it generate?

→ **Example: A medical assistant agent that reads X-rays, summarizes findings, and speaks results.**

Step 2 》 DESIGN STRUCTURED INPUT & OUTPUT

* Use Pydantic AI or JSON Schemas to define what the agent receives and returns.
* Avoid messy text — think like an API.

→ **Tool: Pydantic AI, LangChain Output Parsers**

Step 3 》 PROMPT AND TUNE THE AGENT'S BEHAVIOR

* Start with role-based system prompts
* Use Prompt Tuning or Prefix Tuning for consistent persona and task behavior

→ **Tools: GPT-4, Claude, Prefix Tuning, Prompt Tuning**

Step 4 》 ADD REASONING AND TOOL USE

* Equip the agent with reasoning frameworks:
  ☆ ReAct (Reasoning + Action)
  ☆ Chain-of-Thought
* Allow access to tools like web search, code interpreters, or document retrievers.

→ **Tools: LangChain, OpenAI Tools, ReAct Framework**

Step 5 》 STRUCTURE MULTI-AGENT LOGIC (IF NEEDED)

* Use orchestration frameworks to define agent roles and coordination.
* Create Planner, Researcher, Reporter agents — each with its own input/output schema.

→ **Tools: CrewAI, LangGraph, OpenAI Swarm**

Step 6 》 ADD MEMORY AND LONG-TERM CONTEXT (RAG)

* Does your agent need to remember what happened earlier?
* Use conversational memory, summary memory, or vector-based memory.

→ **Tools: Zep, LangChain Memory, ChromaDB, FAISS**

Step 7 》 ADD VOICE OR VISION CAPABILITIES (OPTIONAL)

* Text-to-speech: Use Coqui or ElevenLabs
* Image understanding: Use GPT-4o or LLaMA 3.2 Vision

→ **Let your agent see and speak.**

Step 8 》 DELIVER THE OUTPUT

* Format outputs into Markdown → PDF or JSON
* Output must be readable and parsable

→ **Tools: Pydantic AI, LangChain Parsers**

Step 9 》 WRAP IN A UI

* Create a front-end
* Use Gradio, Streamlit, or FastAPI

→ **This is what turns your agent into a product.**

Step 10 》 EVALUATE AND MONITOR

* Run test prompts and toolchains to check reliability.
* Use logs, benchmarks, and feedback to improve over time.