PROPOSAL

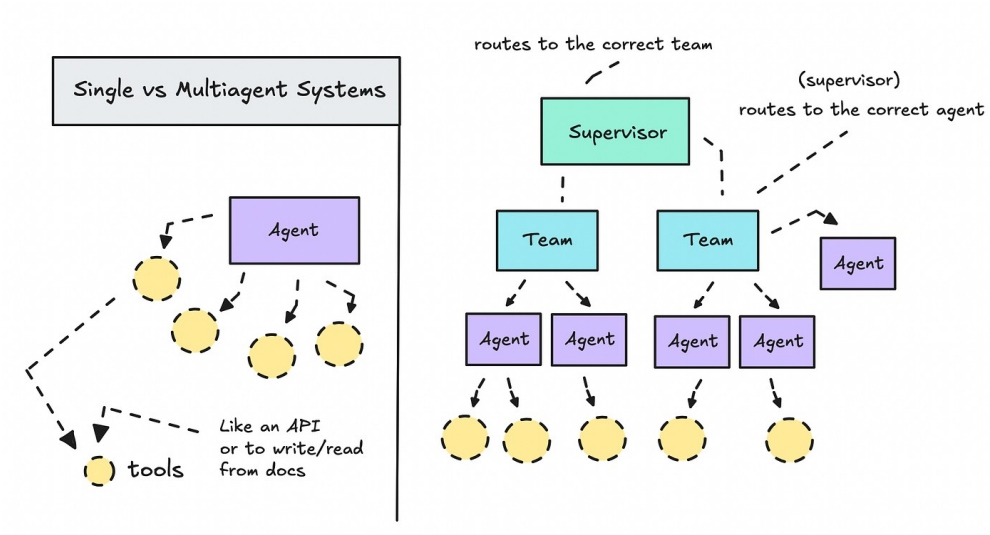
المهارات الأساسية لإتقان الوكلاء الأذكياء والأنظمة الذكية

— The Essential Skills for Mastering AI Agents & Intelligent Systems

دليل تعلم شامل | 2026

PREPARED BY	DATE	STATUS
الخليل العبدالي | إعداد خصيصًا لمسيرتك التدريبية	2026-07-22	Confidential

Introduction | مقدمة



شكل 1: مقارنة بين الوكيل الفردي والوكلاء المتعددين — *Agentic AI: Single vs Multi-Agent Systems*

هذا الدليل الشامل يضع بين يديك خريطة طريق متكاملة للمهارات التي تحتاجها لإتقان بناء وتطوير الوكلاء الأذكياء (AI Agents) والأنظمة الذكية. تم إعداده بناءً على أحدث المصادر لعام 2026، ليكون دليلك العملي في رحلتك من الأساسيات إلى الإنتاج.

يعمل الخبير في هذا المجال على تصميم أنظمة تجمع بين نماذج اللغة الكبيرة (LLMs)، وقواعد المعرفة، وأدوات خارجية، ومنطق تنسيق متقدم — لبناء وكلاء قادرين على التفكير واتخاذ القرارات وتنفيذ المهام بشكل مستقل.

> **INFO** خريطة المهارات في هذا الدليل تغطي 7 محاور رئيسية: الأساسيات → هندسة الأوامر → الاسترجاع والمعرفة → تصميم الوكلاء → الأطر والتنسيق → البروتوكولات والتكامل → الإنتاج والنشر

1. الأساسيات البرمجية — Programming & LLM Foundations

قبل بناء أي وكيل ذكي، لا بد من إتقان الطبقة الأساسية. هذه المهارات هي الركيزة التي يقوم عليها كل ما يليها.

1.1 لغة بايثون ومبادئ البرمجة

بايثون (Python) — اللغة الأساسية: دوال، كلاسات، Decorators, Generators, Async/Await

- Git و GitHub — إدارة الإصدارات والتعاون على الكود
- REST APIs — فهم HTTP، JSON، توثيق واجهات API
- مبادئ تصميم الأنظمة: Microservices, Containerization (Docker)

1.2 أساسيات نماذج اللغة الكبيرة (LLMs)

- فهم كيفية عمل LLMs: Tokenization, Attention, Transformers
- مفاهيم: Temperature, Top-P, Max Tokens, Stop Sequences
- التعامل مع API الموديلات: OpenAI, Anthropic, Google Gemini, Open Source
- بنية الـ Context Window وكيفية إدارتها

1.3 Function Calling — جوهر الوكلاء

Function Calling هو الآلية التي تسمح للنموذج باستدعاء أدوات خارجية. بدونه لا يوجد وكيل ذكي حقيقي. يتضمن تعريف Schemas للأدوات ومطابقة الاستدعاء مع النتائج.

> **TIP** هذه الطبقة تستغرق 2-4 أسابيع لمن لديه خلفية برمجية، و4-8 أسابيع للمبتدئين.

2. هندسة الأوامر المتقدمة — Advanced Prompt Engineering

لم تعد هندسة الأوامر مجرد كتابة نص جيد — أصبحت علمًا دقيقًا يجمع بنية التعليمات المنهجية مع تقنيات التحكم في المخرجات.

2.1 تقنيات الهندسة الأساسية

التقنية	الوصف	الاستخدام
Few-Shot Prompting	إعطاء أمثلة في السياق	تحسين دقة الإجابات في مهام محددة
Chain-of-Thought	طلب تفكير خطوة بخطوة	المسائل المنطقية والحسابية
Structured Output	تحديد JSON Schema للإخراج	ضمان تنسيق مخرجات قابل للبرمجة
System Prompt Engineering	كتابة تعليمات النظام	تحديد الشخصية والقيود والسياق

2.2 هندسة الأوامر للوكلاء

- كتابة تعليمات نظام للوكيل: السياق، الأدوات، القيود، أسلوب العمل
- إدارة السياق: Dynamic Context Window, Sliding Window, Summarization
- Structured Output: JSON Mode, Pydantic Output Parsers
- إدارة الحلقات: Loop Engineering — كتابة مدققات (Verifiers) وليس أوامر

WARNING مبدأ مهم في 2026: Write Verifiers, Not Prompts — أي اكتب مدققات تتحقق من المخرجات بدلاً من محاولة ضبط الأوامر

3. الاسترجاع المعزز وقواعد المعرفة — RAG & Knowledge Systems

RAG هو المهارة الأكثر طلبًا في هندسة الذكاء الاصطناعي لعام 2026. تمكن الوكلاء من الوصول إلى معرفة خارجية دقيقة بدلاً من الاعتماد على التدريب المسبق فقط.

3.1 أساسيات RAG

التقطيع (Chunking): تقسيم النصوص إلى أجزاء مناسبة — Fixed, Semantic, Recursive

- التضمين (Embeddings): تحويل النصوص إلى متجهات باستخدام نماذج التضمين
- قواعد المتجهات (Vector Databases): Pinecone, ChromaDB, Qdrant, Weaviate
- الاسترجاع (Retrieval): Similarity Search, Hybrid Search, Re-Ranking

3.2 تقنيات RAG المتقدمة

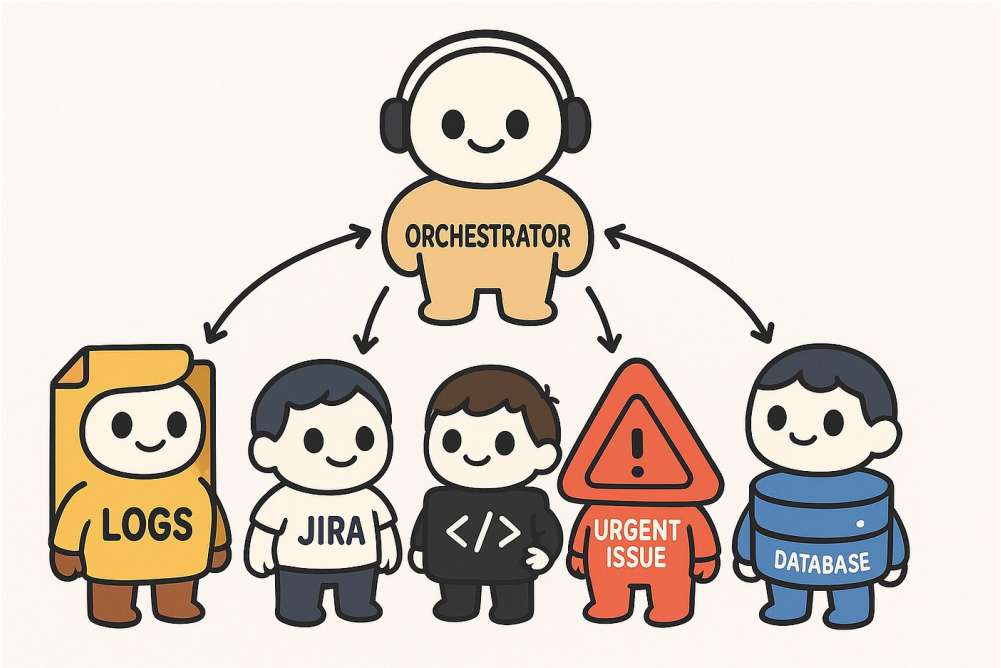
- RAG Fusion: دمج نتائج استعلامات متعددة
- Agentic RAG: الوكيل نفسه يقرر متى وكيف يسترجع
- Graph RAG: استخدام Knowledge Graphs لتحسين الاسترجاع
- Multi-Hop RAG: استرجاع عبر خطوات متعددة وربط المعلومات

TIP خبراء RAG المتمرسون يمكنهم تحسين دقة الإجابات من 50% إلى أكثر من 95% من خلال اختيار استراتيجية التقطيع والتضمين والاسترجاع الصحيحة.

4. تصميم الوكلاء الذكيين — AI Agent Architecture & Design

هذا هو المحور الأهم — فهم بنية الوكيل الذكي. الوكيل ليس شات بوت. إنه نظام متكامل يجمع بين النموذج والأدوات والذاكرة والتخطيط.

4.1 Agent Loop — دورة حياة الوكيل



شكل 2: بنية نظام وكيل ذكي قابل للتوسع — *Scalable Multi-Agent System Architecture*

الوكيل الذكي يعمل في حلقة مستمرة: Perception (استقبال المدخلات) → Reasoning (تفكير) → Action (فعل) → Observation (ملاحظة النتائج) → تكرار حتى اكتمال الهدف.

4.2 مكونات الوكيل الأساسية

المكون	الوظيفة	تقنيات شائعة
النموذج الأساسي (LLM)	دماغ الوكيل — التفكير واتخاذ القرارات	GPT-4o, Claude, Gemini, Llama
الأدوات (Tools)	قدرات الوكيل — واجهات API, DB, Browser	Function Calling, MCP Tools, Custom APIs
الذاكرة (Memory)	تذكر السياق والتجارب	Short-term, Long-term,

	السابقة	Episodic, Semantic
التخطيط (Planning)	تقسيم الأهداف الكبيرة إلى خطوات	ReAct, Plan-and-Execute, Tree-of-Thoughts
التقييم (Reflection)	مراجعة المخرجات وتصحيح الأخطاء	Self-Critique, Loop Engineering

4.3 أنواع أنظمة الذاكرة

- الذاكرة قصيرة المدى (Short-term): السياق الحالي للحوار
- الذاكرة طويلة المدى (Long-term): قاعدة بيانات خارجية للتجارب السابقة
- الذاكرة العرضية (Episodic Memory): تذكر أحداث معينة وحوارات سابقة
- الذاكرة الدلالية (Semantic Memory): المعرفة العامة والحقائق
- الذاكرة الإجرائية (Procedural Memory): كيفية تنفيذ المهام

4.4 أنماط Workflow للوكلاء

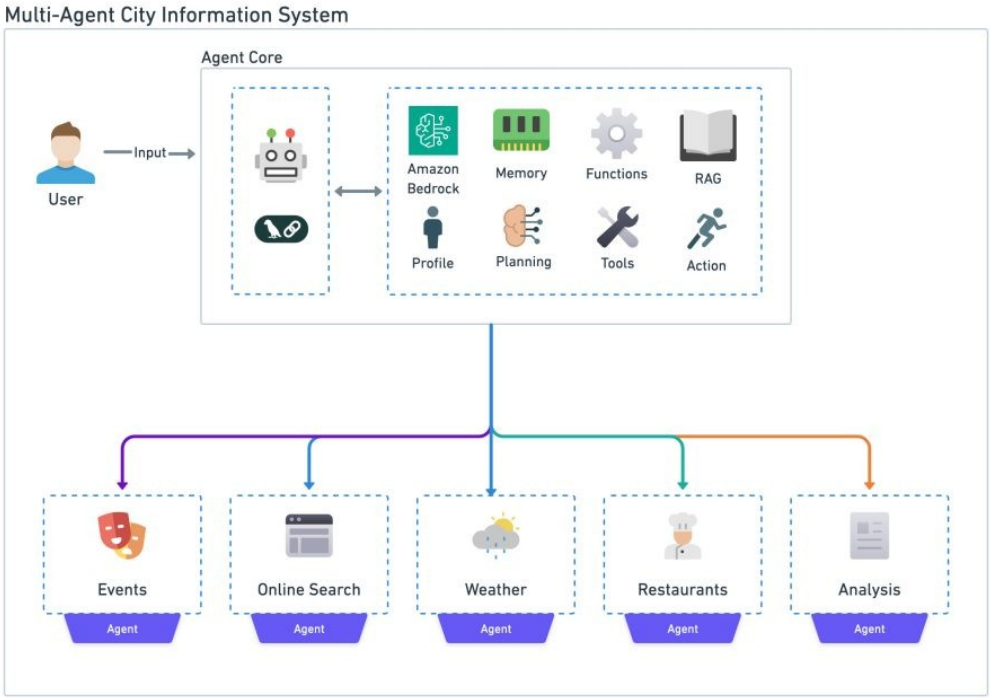
- Prompt Chaining: تنفيذ خطوات متسلسلة — خرج الأولى مدخل الثانية
- Routing: تصنيف المدخل وتوجيهه إلى مسار متخصص
- Parallelization: تنفيذ مهام متزامنة ثم دمج النتائج
- Orchestrator-Workers: وكيل رئيسي يوزع المهام على وكلاء فرعيين
- Evaluator-Optimizer: حلقة تحسين مستمرة بين منشئ ومقيم

WARNING هذا هو الفاصل الحقيقي بين مطور Prompt عادي ومهندس وكلاء محترف. فهم Agent Architecture يحدد قدرتك على بناء أنظمة إنتاجية.

5. أطر العمل والتنسيق بين الوكلاء — Frameworks & Multi-Agent Orchestration

أطر العمل الحديثة توفر بنية تحتية جاهزة للوكلاء: إدارة السياق، الأدوات، تنسيق الوكلاء المتعددين، المراقبة، والتقييم.

5.1 مقارنة أطر العمل الرئيسية في 2026



شكل 3: بناء وكيل متعدد باستخدام *LangGraph — Multi-Agent System with LangGraph*

الإطار	الاستخدام الأفضل	نقاط القوة	نقاط الضعف
LangChain	النماذج الأولية السريعة	تعدد مقدمي النماذج، أدوات جاهزة	المرونة تؤدي للتعقيد
LangGraph	الأنظمة المعقدة ذات الحالة	State Machines، Human-in-the-Loop	منحنى تعلم حاد
CrewAI	الأنظمة متعددة الوكلاء السريعة	نموذج ذهني بديهي Roles-Tasks-Crew	تحكم أقل من LangGraph
AutoGen (Microsoft)	التطبيقات الحوارية المتقدمة	تنسيق المحادثات، Code Execution	تعقيد في الأنظمة الكبيرة
OpenAI Agents SDK	التكامل مع OpenAI	سهولة، توثيق ممتاز	ارتباط بمزود واحد

مقيد بـ GCP	تكامل مع Vertex AI, Gemini	بيئة Google Cloud	Google ADK
أقل نضجًا في الوكلاء	متقدم، إدارة RAG وثائق	التطبيقات كثيفة البيانات	LlamaIndex

5.2 مفاهيم التنسيق المتقدم

- Multi-Agent Systems: تنسيق وكلاء متعددين لأدوار مختلفة
- Supervisor/Orchestrator: وكيل مشرف يدير وكلاء متخصصين
- Debate and Consensus: وكلاء يناقشون ويتوصلون إلى إجماع
- Hierarchical Agents: هيكل هرمي للوكلاء بمستويات صلاحيات
- Checkpointing and State Persistence: حفظ حالة الوكيل لاستئناف العمل

> **INFO** اختيار الإطار المناسب يعتمد على: تعقيد المهمة، الحاجة إلى Human-in-the-Loop، البيئة التقنية للمؤسسة، وميزانية التشغيل.

6. البروتوكولات وتكامل الأدوات — MCP, A2A and API Integration

في 2026، أصبح MCP (Model Context Protocol) و A2A (Agent-to-Agent) هما المعياران الأساسيان لتوصيل الوكلاء بالعالم الخارجي.

6.1 Model Context Protocol (MCP)

MCP هو بروتوكول مفتوح من Anthropic لتوحيد كيفية تواصل الوكلاء مع الأدوات ومصادر البيانات. يشبه USB-C للوكلاء — اتصال واحد موحد للجميع.

- MCP Server: بناء سيرفرات تكشف أدوات ومصادر بيانات
- MCP Client: الاتصال بالسيرفرات من داخل الوكيل
- Stdio Transport: للتطبيقات المحلية والنصوص
- SSE/HTTP Transport: للاتصالات عن بعد والسحابية
- MCP Security: التحقق من الهوية، الصلاحيات، حماية المدخلات

6.2 Agent-to-Agent Protocol (A2A)

بروتوكول من Google يتيح للوكلاء من مزودين مختلفين التواصل مع بعضهم البعض مباشرة — بدون وسيط مركزي. يفتح آفاقًا جديدة للأنظمة الموزعة.

6.3 أدوات التكامل الأخرى

- Agent Communication Protocol (ACP): بروتوكول التواصل بين الوكلاء من Anthropic
- Webhooks and WebSocket: اتصال فوري بين الأنظمة
- API Gateway: إدارة وتقييد الوصول إلى APIs
- Tool Registry: سجل مركزي للأدوات المتاحة للوكيل

7. الإنتاج والنشر والمراقبة — Production, Deployment and LLMOps

بناء وكيل في بيئة تطوير شيء ونشره في الإنتاج شيء آخر تمامًا. هذه المهارات تميز المهندس المحترف.

7.1 النشر والتشغيل

شكل 4: إدارة مسؤولة للأنظمة متعددة الوكلاء — *Responsible Multi-Agent Systems Management*

- FastAPI: بناء APIs لتشغيل الوكلاء
- Docker: حزم وتوزيع الوكيل كحاوية
- Cloud Deployment: AWS, Azure, GCP, أو الخوادم المحلية
- Serverless: نشر بتكلفة منخفضة (AWS Lambda, Cloudflare Workers)

7.2 المراقبة والتقييم (Observability and Evaluation)

الاستخدام	الغرض	الأداة والتقنية
تصحيح الأخطاء، تحليل الأداء	مراقبة وتتبع خطوات الوكيل	LangSmith
فهم سلوك الوكيل وتحسينه	تسجيل كل خطوة من خطوات الوكيل	Tracing and Logging
ضمان الجودة قبل النشر	اختبار الوكيل ضد معايير محددة	Automated Evaluation

Prompt Monitoring	مراقبة جودة الأوامر بمرور الوقت	اكتشاف تدهور الأداء

7.3 الأمان والسلامة

- Prompt Injection Defense: منع اختراق تعليمات النظام
- Input/Output Guardrails: فلاتر إدخال وإخراج
- Sandboxed Execution: تشغيل الكود في بيئة معزولة
- Rate Limiting and Budget Controls: منع الاستخدام المفرط
- Data Privacy: ضمان عدم تسرب البيانات

WARNING الأمان في الوكلاء الذكيين ليس ميزة إضافية — إنه شرط أساسي للنشر الإنتاجي. 70% من فشل مشاريع الوكلاء يعود لقلة الاهتمام بالأمان والمراقبة.

8. خريطة الطريق المقترحة — Suggested Learning Roadmap

هذه خريطة طريق متدرجة تأخذك من الصفر إلى الإتقان، بناءً على خبرتك الحالية في التقنية والذكاء الاصطناعي.

المرحلة الأولى: الأساسيات (2-4 أسابيع)

- تطوير بايثون متقدم: Async/Await, Classes, API Calls
- أساسيات LLMs و API Calls
- جوهر الوكلاء — Function Calling
- مشروع: بناء وكيل بسيط يستقبل الأوامر عبر API

المرحلة الثانية: الاسترجاع والمعرفة (3-5 أسابيع)

- RAG Full Stack: Chunking إلى Embeddings إلى Vector DB إلى Retrieval
- تقنيات التقطيع المتقدمة والتضمين
- RAG مع تحسين الدقة: Re-Ranking, Hybrid Search
- مشروع: نظام RAG على وثائق مؤسستك

المرحلة الثالثة: تصميم الوكلاء (4-6 أسابيع)

- بناء Agent Loop من الصفر (60 سطر كود)
- تكامل الأدوات والذاكرة
- أنماط Workflow: Chaining, Routing, Orchestrator-Workers
- إطار عمل واحد متقدم (LangGraph أو CrewAI)
- مشروع: وكيل بحث متعدد المصادر مع فريق من الوكلاء الفرعيين

المرحلة الرابعة: الإنتاج والاحتراف (3-5 أسابيع)

- MCP Servers: بناء وتكامل
- النشر: FastAPI + Docker + Cloud
- المراقبة: LangSmith, Tracing, Evaluation
- أمان الوكلاء: Guardrails, Sandboxing
- مشروع ختامي: وكيل مؤسسي متكامل مع MCP, RAG, Multi-Agent, ولوحة مراقبة

TIP المدة الإجمالية التقريبية: 12-20 أسبوعًا (3-5 أشهر) بوتيرة دراسة منتظمة. يمكن

تقليصها إلى 8-12 أسبوعًا مع التفرغ الكامل.

9. جدول الأدوات والمصادر — Tools and Resources Reference

الجدول التالي يلخص أهم الأدوات التي يجب إتقانها في كل مجال:

المجال	الأدوات الأساسية	أدوات متقدمة	مصادر تعلم
البرمجة	Python, Git, REST APIs	Docker, FastAPI, Async	Real Python, FastAPI Docs
LLMs	OpenAI API, Anthropic API	Vertex AI, Ollama, vLLM	OpenAI Docs, DeepLearning.AI
Prompt Engineering	System Prompts, Chain-of-Thought	Loop Engineering, Verifiers	Anthropic Docs, PromptLayer
RAG	ChromaDB, LangChain	Graph RAG, Re-Ranking, Hybrid	LangChain Docs, Pinecone Docs
Agent Frameworks	LangGraph, CrewAI	AutoGen, Google ADK, OpenAI SDK	LangChain Academy, CrewAI Docs
MCP	MCP SDK (Python)	MCP Security, WebMCP	MCP Docs, modelcontextprotocol.io
المراقبة	LangSmith	Arize, Braintrust, Datadog	LangSmith Docs
النشر	FastAPI, Docker	Kubernetes, Serverless	Docker Docs, FastAPI Tutorial
الأمان	Input Validation	Guardrails, Sandboxing	OWASP LLM Top 10

10. توصيات ختامية — Final Recommendations

بناءً على تحليل سوق العمل في 2026 واحتياجات المؤسسات للوكلاء الأذكياء، إليك أهم التوصيات لرحلتك:

1. ابدأ بالمشاريع لا النظريات — كل مشروع تبنيه يعلمك أكثر من أي دورة
2. RAG هو المهارة الأكثر طلبًا وظيفيًا — اجعله أولوية بعد الأساسيات
3. أتقن إطار عمل واحد بعمق قبل الانتقال لآخر — LangGraph أو CrewAI
4. تعلم بناء Agents من الصفر قبل استخدام الأطر — تفهم الجوهر
5. MCP أصبح المعيار الصناعي لتكامل الأدوات — لا تتجاهله
6. المراقبة والتقييم (Observability) تفصل بين الهاوي والمحترف
7. الأمان ليس اختياريًا — كل وكيل في الإنتاج يحتاج Guardrails
8. تخصص في مجال عمودي: Legal Tech, HealthTech, FinTech, Education
9. تذكر دائمًا: Write Verifiers, Not Prompts

> **INFO** المفتاح ليس في معرفة كل أداة، بل في فهم متى ولماذا تستخدم كل منها — System Design Thinking هو ما يصنع مهندس الوكلاء المحترف.

تم إعداد هذا الدليل بواسطة سنا — المساعد الذكي على منصة Sana AI Agent Platform، بإشراف وتطوير مبحث للبحث والتطوير (Mabahith for Research and Development).
للتواصل: https://sana.com | info@sanaabot.com