

Welcome to Day 3! 🚀

Agentic AI: AI Agents That Think and Act

Today's Journey:

- ✓ From Chatbots → Intelligent Agents
- ✓ Function Calling & Tool Use
- ✓ Multi-step Reasoning
- ✓ Build Agents That Can Act!

AI That Can Think, Plan, and Execute! 🧠⚡

Quick Recap: Days 1 & 2



Day 1: Basics

- API calls to LLMs
- Prompt engineering
- Chatbots with memory

Day 2: Automation

- n8n workflows
- Visual automation
- Service integrations

Today: The Next Level!

We've built chatbots that respond. Now let's build agents that can **think, plan, and act** using tools!

What is an AI Agent?

An AI that can reason, plan, and use tools to accomplish tasks

Chatbot (Day 1)



- Responds to questions
- Uses training data
- No actions/tools
- Reactive

AI Agent (Day 3)

- Plans multi-step tasks
- Uses tools/functions
- Makes decisions
- Proactive & autonomous

💡 Key Difference: Agents can **do things**, not just talk about them! They can search, calculate, create files, call APIs, and more.

Why Agents Matter in Business



The AI evolution and its business impact

The AI Evolution



- **2022:** Chatbots
(answer questions)
- **2023:** Advanced chatbots (better conversations)
- **2024-2025:** Agents (take action)
- **2026:** Agentic AI everywhere ← **We are here**

Business Impact Comparison

Before Agents (Chatbots only):

Customer: "What's my order status?"
Bot: "Let me help! Please log into your account and check..."
Result: Customer still has to do the work

With Agents:

Customer: "What's my order status?"
Agent: Looks up order in database → "Your order #12345 shipped yesterday and will arrive Tuesday. Here's tracking: [link]"
Result: Task completed automatically

Key Metrics

- ✓ Traditional chatbots:
30-40% of queries
end-to-end
- ✓ AI Agents: **70-80%**
of queries end-to-end
- ✓ Cost reduction: **60%**
fewer escalations to
humans
- ✓ Customer
satisfaction: **45%**
improvement

Real Business Value

- ✓ **Autonomous execution** -
Complete tasks
without human
intervention
- ✓ **24/7 operation**
- Work while you
sleep
- ✓ **Scale infinitely**
- Handle 1,000x
volume with same
cost
- ✓ **Reduce human workload** - Team
focuses on complex
problems only

Real-World Agent Examples



AutoGPT / BabyAGI

Autonomous agents that can plan and execute complex tasks across multiple steps

GitHub Copilot Agents

AI coding assistants that can read code, search docs, and write code

Customer Service Agents

Agents that can look up orders, process refunds, and update databases

Research Agents

Agents that search, analyze, and summarize information from multiple sources

Common Pattern:

- User asks a question or gives a task
- Agent plans what to do
- Agent uses tools to gather information
- Agent reasons about the results

- Agent provides answer or completes task

Agent vs Chatbot: The Business Case 💰

Comparing costs and ROI across three approaches

Cost Comparison Table

Approach	Cost per Interaction	Resolution Time	Escalation Rate	Availability
Traditional Support	\$25-45	10-30 minutes	60% need human	Business hours only
Chatbot (Day 1)	\$0.50-2	2-5 minutes	60% need human	24/7
AI Agent (Day 3)	\$0.10-0.50	30 sec - 2 min	20% need human	24/7

Savings:

- Chatbot vs Human:
90%
- Agent vs Human:
98%
- Agent vs Chatbot:
75%

Real Example - E-commerce:

10,000 tickets/month:
Human: \$350K/mo → Chatbot: \$150K/mo (40% savings)
→ Agent: **\$75K/mo (78% savings)**

Why Agents Win:

- Can access databases
- Can update systems
- Can complete transactions
- Can make decisions based on rules
- Handles 80% of requests end-to-end

Core Agent Concepts



1. Function Calling / Tool Use

AI can request to call functions you define. Like giving AI superpowers!

- Calculator functions
- Web search
- File operations
- API calls

2. Planning

Agent breaks down complex tasks into steps

- Analyzes the request
- Decides what tools to use
- Plans the sequence

3. Execution

Agent calls tools and processes results

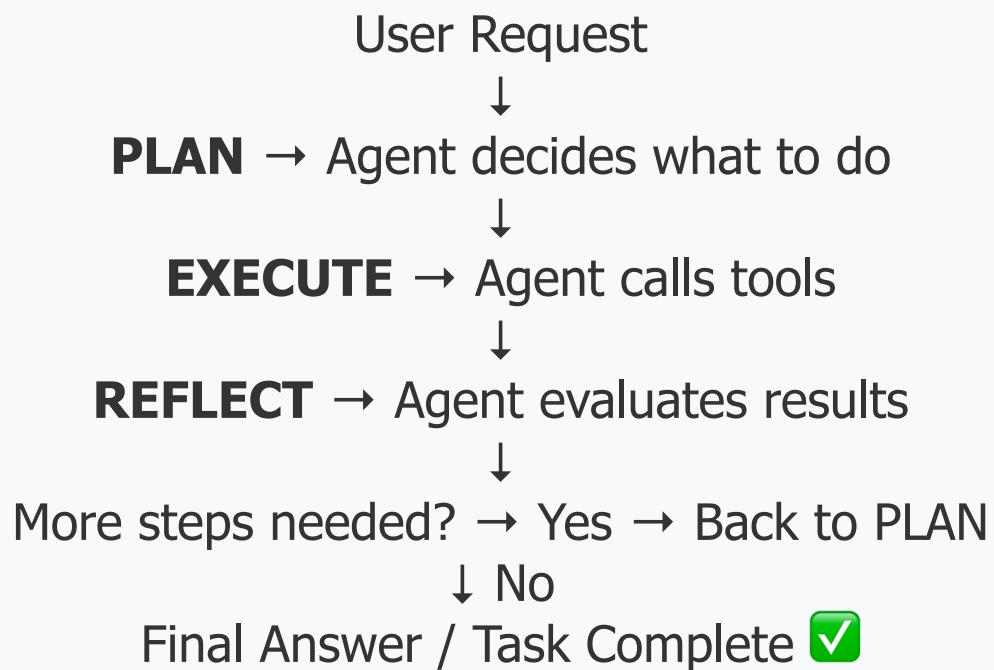
- Calls functions
- Gets results
- Processes information

4. Reflection

Agent evaluates progress and adjusts

- Checks if task is complete
- Decides if more steps needed
- Handles errors

The Agent Loop



This loop continues until: The task is complete, maximum iterations reached, or an error occurs

The Agent Loop in Real Business Scenarios



Understanding how the agent loop works in practice

Scenario 1: Customer Return Request

User Request: "I want to return my order"

PLAN: Agent thinks:

- Need order details
- Check return policy
- Verify eligibility
- Process if approved

EXECUTE: Agent actions:

- Look up order in database
- Check purchase date (within 30 days?)
- Check item condition
- Calculate refund amount

REFLECT: Agent evaluates:

Scenario 2: Research Competitive Pricing

User Request: "What are competitors charging?"

Iteration 1:

- **PLAN:** Search for competitor products
- **EXECUTE:** Web search for prices
- **REFLECT:** Found 2 competitors, need more
- More steps? Yes → Continue

Iteration 2:

- **PLAN:** Search 3 more competitors
- **EXECUTE:** Additional searches, check promotions

- Order eligible ✓
- Customer verified ✓
- Refund calculated ✓

Result: Refund approved and processed in 45 seconds

- REFLECT: Now have complete data ✓
- More steps? No → Present analysis

Result: Comprehensive pricing analysis in 2 minutes

Why This Matters:

Without Agent Loop:

- Human manually searches each competitor
- Takes 2-3 hours
- Might miss competitors
- Data gets stale quickly

With Agent Loop:

- Automated multi-step process
- Completes in 2 minutes
- Comprehensive coverage
- Can run daily for up-to-date data

Business Impact: 90x faster execution • 100% consistency • Scales to 1000s of products

Function Calling: The Foundation



How It Works:

1. You define functions/tools (what AI can do)
2. You describe them to the AI (name, parameters, description)
3. AI decides when to call them
4. You execute the function
5. You send result back to AI
6. AI uses result to respond

Example Function Definition:

```
{  
  "name": "get_weather",  
  "description": "Get weather for a location",  
  "parameters": {  
    "type": "object",  
    "properties": {  
      "location": {"type": "string"}  
    }  
  }  
}
```

Exercise: Personal Assistant Agent



What You'll Build:

An AI agent that can **book meetings** and **send emails** by integrating with n8n workflows!

Capabilities:

- Book meetings (creates real Google Calendar events)
- Send emails (sends real emails via Gmail)
- Extracts intent and entities from natural language
- Multi-step reasoning (asks for missing info)
- Integrates with n8n workflows

File:

```
personal_assistant_agent_starter.py
```

90 minutes

Prerequisites:

- Day 1 & 2 completed
- n8n workflow set up
- Gmail & Calendar credentials

 **This is a real-world agent!** It actually creates calendar events and sends emails.
You'll see real results!

Personal Assistant: How It Works



User: "Book meeting with John about falling stats, email John@gmail.com"



Agent (Python):

- Extracts intent: "book_meeting"
- Extracts entities: email, topic, time?
- Asks: "What time?" (if missing)



Agent calls n8n webhook:

- Sends structured data



n8n Workflow:

- Creates Google Calendar event
 - Sends email invite



Agent responds:

"Done! Meeting scheduled for Tuesday 2pm" A green square icon with a white checkmark inside, indicating success.

Agent Responsibilities:

- Intent extraction
(meeting vs email)
- Entity extraction
(emails, names, topics)
- Ask for missing information
- Call n8n with structured data

n8n Responsibilities:

- Receive structured data
- Route by action type
- Create calendar events
- Send emails

Personal Assistant: Step-by-Step



Step 1: Set Up n8n Workflow (15 min)

- Import personal_assistant_workflow.json
- Configure Google Calendar OAuth2 credential
- Configure Gmail OAuth2 credential
- Activate workflow and copy webhook URL

Step 2: Configure Agent Code (5 min)

- Add N8N_WEBHOOK_URL to .env file
- Install: pip install requests

Step 3: Build the Agent (60 min)

- Define functions for intent extraction
- Implement `trigger_n8n_webhook()` function
- Complete agent loop with function calling
- Handle missing information

Step 4: Test and Iterate (10 min)

- Try booking meetings
- Try sending emails
- See real calendar events and emails created!

Personal Assistant: Key Functions



1. Extract Meeting Intent

```
{ "name": "extract_meeting_intent", "description":  
  "Extract meeting booking details...", "parameters":  
  { "attendee_email": "string", "attendee_name":  
    "string", "topic": "string", "preferred_time":  
    "string", "duration_minutes": "integer" } }
```

2. Extract Email Intent

```
{ "name": "extract_email_intent", "description":  
  "Extract email sending details...", "parameters":  
  { "recipient_email": "string", "subject": "string",  
    "message": "string" } }
```

3. Trigger n8n Webhook

```
trigger_n8n_webhook(action="book_meeting", data={  
    "attendee_email": "john@example.com", "topic":  
    "falling stats", "preferred_time": "Tuesday 2pm",  
    ... })
```

Personal Assistant: Example Flow



Example 1: Book Meeting

User: "Book the meeting with John about his falling stats, his email is John@gmail.com"

Agent: "What time would you like to schedule this meeting?"

User: "Tuesday 2pm"

Agent: [Calls n8n webhook]

Agent: "Done! Meeting scheduled for Tuesday at 2pm. Calendar invite sent to John@gmail.com."

Example 2: Send Email

User: "Send an email to sarah@example.com about the meeting tomorrow"

Agent: "What should the subject be? And what message?"

User: "Subject: Meeting Tomorrow, Message: Don't forget our meeting at 2pm"

Agent: [Calls n8n webhook]

Agent: "Email sent successfully to sarah@example.com!"

What Makes This Agentic:

- Agent extracts intent automatically
- Agent asks for missing information
- Agent calls n8n to execute actions

- ✓ Real results: actual calendar events and emails!

Decision Framework: When to Build AI Agents



A practical guide for business decision-making

Build an Agent When

-  Task Requires Multiple Steps (3+)
-  Need to Use Multiple Tools (2+)
-  Decision-Making Required
-  Task Needs to Complete End-to-End
-  High Volume + Repetitive (100+/day)

DON'T Build an Agent When

-  Needs Human Judgment
-  Simple Single-Step Task
-  Requires Creativity
-  High-Stakes Decisions
-  Constantly Changing Rules

The ROI Calculation: Process Return Requests

Volume: 200/day × 250 days = 50,000/year • Current: \$6.25/request = **\$312,500/year**

Agent build: \$10K • Operation: \$5K/year • **ROI: 20x in first year** 

Factor	Chatbot	Agent
Steps	1-2	3+
Tools Needed	0-1	2+
Cost to Build	\$2K-5K	\$5K-15K
ROI Timeline	Immediate	1-6 months

Common Agent Patterns



1. Single Function Call

User asks → Agent calls function → Returns result

Example: Calculator agent

2. Multi-Step Loop

User asks → Agent plans → Executes → Reflects → Repeats if needed

Example: Research agent

3. Parallel Execution

Agent calls multiple functions simultaneously

Advanced pattern

4. Error Recovery

Agent handles errors and tries alternative approaches

Production-ready pattern

Real-World Agent Applications



Personal Assistants

- Schedule meetings
- Send emails
- Manage tasks
- Research topics

Customer Service

- Look up orders
- Process refunds
- Update databases
- Escalate issues

Code Generation

- Read codebases
- Search documentation
- Write code
- Test and debug

Data Analysis

- Query databases
- Process data
- Generate reports
- Create visualizations

Planning Your Agent Project: Step-by-Step



From idea to production in 10 weeks

Phase 1: Define (Week 1)

- Identify the task & current process
- Map the workflow & decision points
- Define success metrics

Phase 2: Design (Week 2)

- List required tools & systems
- Define decision logic & rules
- Design agent flow

Phase 3: Build MVP (Week 3-4)

- Start with ONE tool first
- Test happy path, errors, edge cases
- Get real user feedback early

Phase 4: Measure (Ongoing)

- Track: automation rate, time, accuracy
- Iterate: Launch → Analyze → Fix → Expand

Timeline (10 weeks):

- Week 1: Planning
- Week 2-3: Build MVP
- Week 4: Internal testing
- Week 5: Beta (10 users)
- Week 6-8: Iterate
- Week 9: Full rollout
- Week 10+: Monitor

Budget:

- Planning & Design:
\$3K-5K
- Development: **\$5K-15K**
- Testing & Iteration:
\$2K-5K
- Maintenance: **\$1K-2K/month**

Agent Frameworks (Overview) 🛠

Popular Frameworks:

- **LangChain** - Popular Python framework for building agents
- **LlamaIndex** - Great for RAG and data agents
- **AutoGPT** - Autonomous agent framework
- **BabyAGI** - Task-driven autonomous agent



Today we're building from scratch! This gives you a deep understanding. Later, you can use frameworks to build faster.

When to Use Frameworks:

- Complex multi-agent systems
- Production applications
- Need pre-built components
- Want to move fast

Agent Best Practices



Function Design:

- Clear descriptions
- Well-defined parameters
- Handle errors gracefully
- Return useful results

Safety & Limits:

- Set max iterations
- Validate inputs
- Rate limiting
- Cost monitoring

System Prompts:

- Clear role definition
- Instructions on tool use
- Error handling guidance
- Output format preferences

Testing:

- Test each function
- Test agent loop
- Test error cases
- Monitor costs

Preparing for Day 4: Projects

Project Ideas Using Agents:

-  Personal assistant agent
-  Research & summarization agent
-  Code generation agent
-  Email processing agent
-  Data analysis agent
-  Shopping assistant agent

What You've Learned:

- Function calling basics
- Building agents
- Multi-step reasoning
- Agent + n8n integration
- Real-world automation

Next Steps:

- Choose a project idea
- Plan your agent
- Define your tools
- Build and test!

Agents + n8n = Superpower! ⚡

What You Just Built:

Personal Assistant Agent that integrates with n8n to book meetings and send emails!

Agent (Python Code):

- Intent extraction
(meeting vs email)
- Entity extraction
(emails, names, topics)
- Multi-step reasoning
- Calls n8n webhook

n8n (Visual Workflow):

- Receives structured data
- Routes by action type
- Creates Google Calendar events
- Sends Gmail emails

More Integration Ideas:

-  n8n workflow → Calls agent API
-  Agent processes email → n8n sends notification
-  Agent researches → n8n saves to database
-  Agent analyzes data → n8n generates reports

Best of both worlds! 

Congratulations!



What You've Accomplished:

- Understood what AI agents are
- Learned function calling / tool use
- Built Personal Assistant agent
- Integrated agents with n8n workflows
- Understood agent patterns and loops
- Created real calendar events and emails!

Key Takeaways:

- Agents can think and act
- Function calling enables tool use
- Multi-step reasoning is powerful
- Agents are the future of AI!

Next Steps:

- Review solutions
- Experiment with agents
- Plan Day 4 project
- Build something amazing!

You're now an Agent Builder!



Questions? Let's discuss what you learned and any questions you have!

Why Learning Agents Matters for Your Career



The opportunity window is NOW

Industry Timeline

- **2024-2025:** Early adoption • Skills in high demand
- **2026:** Mainstream
← **We are here**
- **2027-2028:** Standard expectation

New Roles (\$100K-200K):

- AI Agent Engineer: **\$120K-180K**
- Agentic AI PM: **\$110K-160K**
- Agent Architect: **\$150K-200K**

Your Competitive Advantage

You Have:

- Built agents from scratch
- Function calling & tool use
- Multi-step reasoning
- Can design agent systems

Most Graduates:

- Maybe used ChatGPT
- No agent experience
- No practical AI skills

You're 2-3 years ahead!



Career Paths:

- Agent Specialist (startup, high equity)
- Enterprise Builder (stable, high salary)
- Consulting (\$150-300/hr)
- Found Your Company

Action Items:

- **This Week:**
Complete Personal Assistant agent, document in portfolio
- **This Month:** Build 2-3 portfolio projects, blog post, LinkedIn
- **Next 3 Months:**
Apply to AI companies, contribute to open source, network

The Future is Agentic!