

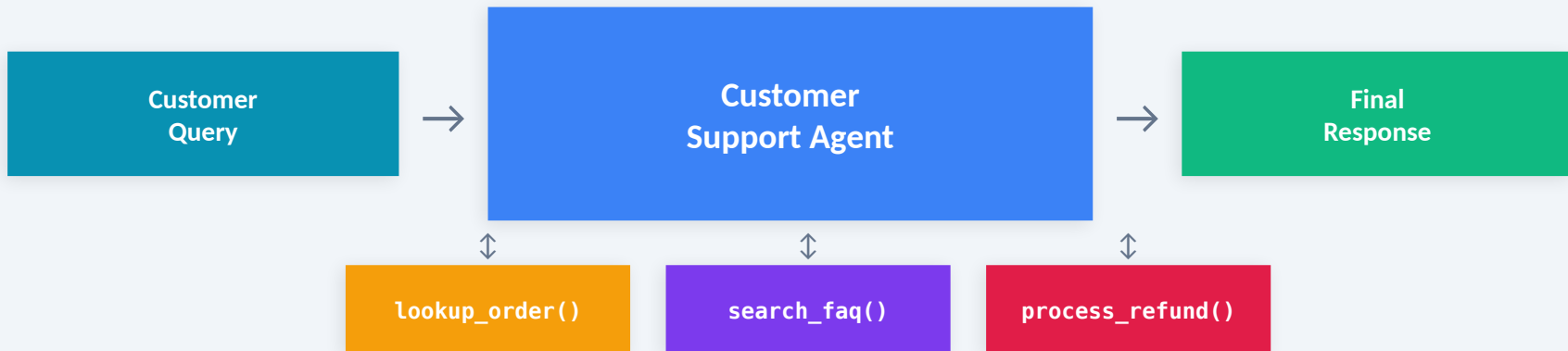
03

Single Agent Pattern

One agent with tools and a reasoning loop — the foundation

Single Agent: Customer Support

One agent with specialized tools and a focused system prompt. Most tasks should start here.



When to Use

- Task requires reasoning + tool use
- A single system prompt covers needed expertise
- Examples: support, research, data retrieval

Context Strategy

One conversation thread — all messages (system, user, assistant, tool) in a single list that accumulates. Simple but can hit token limits on long conversations.

Exercise: Customer Support Agent

exercises/03_single_agent/01_customer_support_agent.py

Agent Definition (using commons/agent.py)

```
agent = Agent(  
    name="Support Agent",  
    system_prompt=SYSTEM_PROMPT,  
    tools=[  
        pydantic_function_tool(LookupOrderParams, ...),  
        pydantic_function_tool(SearchFaqParams, ...),  
        pydantic_function_tool(ProcessRefundParams, ...),
```

```
    ],  
    tool_functions={
```

Multi-Turn Conversation Loop

```
# messages persists across turns  
messages: list[dict] = []  
  
for customer_msg in CUSTOMER_MESSAGES:  
    messages.append(  
        {"role": "user",  
         "content": customer_msg}
```

```
    )  
# run() handles the full agent loop
```

3-Turn Conversation Flow — Messages List Grows

Turn 1	"Help with order ORD-1001"	→ <code>lookup_order()</code>	→ Status: delivered, items listed
Turn 2	"What's your return policy?"	→ <code>search_faq()</code>	→ 30-day return policy explained
Turn 3	"I'd like a refund"	→ <code>process_refund()</code>	→ Refund approved, REF-XXXXX

Context grows: 3 msgs → ~8 msgs → ~14 msgs. All history visible to the agent at each turn.

Chapter 3 Recap: Single Agent Pattern

- 1 The Agent dataclass bundles name, system_prompt, tools, and tool_functions
- 2 run() handles the entire Reason → Act → Observe loop internally
- 3 Persistent messages list = the agent's memory (grows with each turn)
- 4 Start with a single agent — only add complexity when one agent isn't enough