






# What is the **Runner** Class?

In simple terms:

The **Runner** class is the **main engine** that takes an **Agent** and executes it — combining:

-  System instructions
-  User prompt
-  Tools (optional)
-  Context (optional)
-  Memory (optional)

It handles the full **request-response lifecycle** of the agent.

---

## Why Do We Need the Runner Class?

Imagine you've defined an agent:

```
agent = Agent(  
    name="Python Tutor",  
    instructions="You are a helpful tutor who explains Python  
concepts simply.",  
    model="gpt-4"  
)
```

Now what? Just having an agent defined doesn't **do anything** yet.

That's where **Runner** comes in — it says:

“Okay, now that I have an agent, run it with a user prompt and give me a response.”

---

## Real-World Analogy

### Agent = Chef

- Knows how to cook
- Has tools, instructions, and recipes

### Runner = Kitchen Manager




- Accepts the customer's order
- Prepares ingredients
- Calls the chef to cook
- Delivers the meal (response)




Without a **Runner**, the chef (agent) just stands there. The Runner **makes the system go**.

---

## What Does the Runner Actually Do?

Behind the scenes, **Runner** handles:

Step	What Runner Does
 Prepares	Loads the agent's model, tools, memory
 Gathers context	Uses memory/context if any
 Accepts user prompt	The <code>run(prompt)</code> input

 Applies instructions	Adds the system message
 Calls model	Sends messages to the LLM
 Handles output	Parses and returns final response

---

## Key Responsibilities of Runner

Task	Explanation
Run agent logic	Handles the entire conversation cycle
Manage tools	Executes tools if the agent calls them
Support memory	Uses agent memory (if defined) to guide future steps
Enable multi-step reasoning	If the model replies with follow-up tasks, Runner loops
Return final answer	Sends back a clean, formatted response

---

## Example Use:

```
from openai import AssistantAgent, Runner
```

```
agent = AssistantAgent(name="Helper", instructions="You are a  
creative assistant.")
```

```
response = Runner.run(
    agent=agent,
    user_prompt="Write a poem about AI and creativity."
)

print(response)
```

Just one line of `Runner.run()` does all the magic 

---

## Final Concept Summary (Exam Style):

Concept	Meaning
<code>Runner</code>	Core utility class that runs agent logic
Purpose	Execute agent with instructions, user input, tools, memory
Why needed?	Agent alone is just configuration — Runner makes it interactive
Design	<code>@classmethod run()</code> handles single-run logic cleanly
Analogy	Kitchen manager that executes a chef's recipe when an order comes