

What is `Runner.run()`?

In the **OpenAI Agents SDK**, the `Runner` class is responsible for **executing the agent** — meaning:

“Take the user’s message, apply the agent’s instructions and tools, and produce a response.”

The method signature looks like this:

```
@classmethod
def run(cls, user_prompt: str, **kwargs) -> str:
    ...
```

Why is the `user_prompt` passed to `Runner.run()`?

Because:

➤ **It’s the main input from the user.**

Imagine:

```
Runner.run("How does generics work in Python?")
```

This is equivalent to the user typing a message in the chat box.

✓ `user_prompt` = the natural language **query or instruction** you want the agent to process.

It is:

- Sent to the **agent model**
 - Combined with the **system instructions**
 - Processed by tools, memory, etc.
 - Used to generate the final **response**
-

Think of `Runner.run()` as a **conversation engine**:

- `instructions` = job description (e.g. "You're a Python tutor")
- `user_prompt` = the question asked (e.g. "Teach me about generics")
- Output = the tutor's answer

So `Runner.run(user_prompt)` starts the whole chat.





Why is `run()` a classmethod?

➤ **Because you're not working with a specific Runner object.**

Instead, you're saying:

"Use this class (`Runner`) to start a one-time run of the agent logic."

✓ Benefits of using `@classmethod` here:

Reason	Why it Matters
 Simple usage	You don't need to create a Runner instance manually
 Class-level logic	The method can setup/configure behavior from the class
 Clean API	<code>Runner.run(...)</code> is more readable and direct
 SDK Design Pattern	Keeps the runner stateless , reusable, and easy to invoke

Code Analogy: Compare `classmethod` vs `instance method`

```
class Runner:
    @classmethod
    def run(cls, prompt: str):
        print(f"Running with prompt: {prompt}")
```

You can call:

```
Runner.run("Tell me a joke")
```

You don't need:

```
runner = Runner()  
runner.run("...")
```

That's less clean and unnecessary for **single-run logic**.

Summary: Why both are used

Element	Purpose
<code>user_prompt</code>	The user's message — what they want the agent to respond to
<code>Runner.run()</code>	The method that launches the agent logic
<code>@classmethod</code>	Lets you call it from the class itself without creating an instance; ideal for static, reusable workflows

Exam-Ready Takeaway

- `Runner.run(prompt)` is the **main entry point** for using an agent.
- The **user prompt** drives the conversation.
- `@classmethod` is used because you don't need a `Runner` instance — it runs once and returns a result.