□ 100 Q&A on Runner

Runner Basics

1.

   Q: What is a Runner in an Agentic AI system?
   A: A Runner is the execution engine that manages how an agent interacts with the environment, tools, and user.

2.

   Q: What is the primary role of a Runner?
   A: To control the execution flow of conversations and tool usage.

3.

   Q: Which entity decides the maximum number of dialogue turns?
   A: The Runner.

4.

   Q: What is a "turn" in Runner execution?
   A: A single input-output exchange between the agent and the system.

5.

   Q: Can a Runner exist without an Agent?
   A: No, it requires an agent to execute tasks.

6.

   Q: Which parameter ensures execution doesn’t run indefinitely?
   A: max_turns.

7.

   Q: What happens if max_turns is exceeded?
   A: The Runner stops execution.

8.

   Q: Who sets execution policies: Agent or Runner?

A: Runner.

9.

Q: Can a Runner call multiple tools in one turn?
A: Yes, depending on configuration.

10.

Q: What is the default responsibility of a Runner?
A: To orchestrate agent responses and enforce limits.

Execution Control

11.

Q: What is execution control in Runner?
A: The process of managing how agent outputs and tool calls are executed step by step.

12.

Q: What prevents an infinite loop in Runner?
A: Execution limits (max_turns, safety guards).

13.

Q: How does a Runner handle failed tool calls?
A: It retries, logs errors, or stops based on settings.

14.

Q: What is a "step" in Runner execution?
A: One cycle of agent reasoning and action execution.

15.

Q: Can execution control override agent decisions?
A: Yes, via hooks or guardrails.

16.

Q: What happens if execution control is disabled?
A: The agent may run without constraints, risking infinite loops.

17.

Q: Which parameter decides execution length?
A: max_turns.

18.

Q: How does Runner handle structured outputs?
A: By validating agent outputs against schemas.

19.

Q: Can Runner enforce timeouts?
A: Yes, to prevent long-running executions.

20.

Q: What is the difference between max_turns and timeout?
A: max_turns controls steps; timeout controls time duration.

Custom Runner

21.

Q: What is a Custom Runner?
A: A Runner with modified logic for specific agent behaviors.

22.

Q: Why create a custom runner?
A: To enforce unique execution strategies or workflows.

23.

Q: Can custom runners override tool selection?

A: Yes, they can implement custom tool choice.

24.

Q: Which programming concept is used to extend Runner functionality?
A: Subclassing or composition.

25.

Q: Can custom runners log execution differently?
A: Yes, custom logging is supported.

26.

Q: What role does a custom runner play in debugging?
A: It allows detailed trace and flow control.

27.

Q: Can you define custom retry logic in a Runner?
A: Yes, by overriding error handling methods.

28.

Q: How can custom runners improve efficiency?
A: By pruning redundant steps or optimizing tool calls.

29.

Q: Do custom runners support multi-agent orchestration?
A: Yes, with proper coordination logic.

30.

Q: Example use-case of a custom runner?
A: Handling financial trading bots with strict risk limits.

Runner vs Agent

31.

Q: Who generates outputs: Runner or Agent?
A: Agent generates; Runner controls.

32.

Q: Who enforces limits: Runner or Agent?
A: Runner.

33.

Q: Can an Agent work without Runner?
A: Not in most frameworks; execution requires Runner.

34.

Q: Who manages retry policies?
A: Runner.

35.

Q: Is Runner logic domain-specific?
A: It can be generic or customized.

36.

Q: Does Runner have intelligence?
A: No, it enforces execution rules.

37.

Q: Can Runner alter the agent’s reasoning chain?
A: Indirectly, by limiting steps or altering context.

38.

Q: Who maintains dialogue history?
A: Usually the Runner.

39.

Q: Can Runner delegate execution to another runner?
A: Yes, with handoff.

40.

Q: Is Runner replaceable?
A: Yes, custom runners can replace default ones.

Max Turns

41.

Q: What is max_turns in Runner?
A: A parameter defining maximum execution steps.

42.

Q: Default risk if max_turns is too high?
A: Infinite or unnecessary loops.

43.

Q: What if max_turns is set too low?
A: Task may stop before completion.

44.

Q: Does max_turns include tool retries?
A: Yes, retries count as turns.

45.

Q: Can max_turns be dynamic?
A: Yes, adjusted based on task.

46.

Q: Which part of Runner enforces max_turns?
A: The execution loop.

47.

Q: What is a safe default for max_turns?
A: Depends on task; usually 10–20.

48.

Q: Can max_turns be unlimited?
A: Yes, but risky.

49.

Q: Who decides max_turns: agent or developer?
A: Developer via Runner config.

50.

Q: Example of high max_turns use?
A: Multi-turn research tasks.

Error Handling

51.

Q: What if a tool fails in Runner execution?
A: Runner retries or stops.

52.

Q: Does Runner log all errors?
A: Yes, usually.

53.

Q: Can Runner implement fallback logic?
A: Yes, custom fallback paths are possible.

54.

Q: What is the advantage of controlled retries?
A: Prevents repeated failures.

55.

Q: Can error handling be customized?
A: Yes, in custom runners.

56.

Q: How does Runner prevent silent failures?
A: By raising errors or warnings.

57.

Q: Can errors trigger handoff?
A: Yes, Runner can delegate.

58.

Q: Does Runner stop on first error?
A: Not necessarily, depends on config.

59.

Q: Who decides retry count?
A: Runner configuration.

60.

Q: Can retries exceed max_turns?
A: No, they count toward it.

Advanced Runner

61.

Q: Can Runner orchestrate multiple agents?
A: Yes, via coordination.

62.

Q: Can Runner handle asynchronous execution?
A: Yes, advanced runners support async.

63.

Q: What is parallel execution in Runner?
A: Running multiple tasks simultaneously.

64.

Q: Can a Runner pause execution?
A: Yes, via suspension logic.

65.

Q: Can Runner resume later?
A: Yes, if state is saved.

66.

Q: Can Runner track agent confidence scores?
A: Yes, through custom metrics.

67.

Q: Does Runner handle structured outputs?
A: Yes, by validating schemas.

68.

Q: Can Runner inject system prompts dynamically?
A: Yes, via control logic.

69.

Q: Can Runner limit API calls?
A: Yes, through execution constraints.

70.

Q: What is multi-turn orchestration?
A: Coordinating multiple steps across turns.

Debugging & Tracing

71.

Q: How does Runner help in debugging?
A: By logging agent decisions.

72.

Q: Can Runner show intermediate reasoning?
A: Yes, via trace mode.

73.

Q: What is a trace log?
A: A detailed record of execution steps.

74.

Q: Can Runner output structured traces?
A: Yes, in JSON or logs.

75.

Q: Does tracing impact performance?
A: Slightly, due to extra logging.

76.

Q: What is the benefit of tracing in Runner?
A: Easier debugging and transparency.

77.

Q: Can tracing be disabled?
A: Yes, to improve performance.

78.

Q: Can Runner support external monitoring tools?
A: Yes, via hooks.

79.

Q: Is Runner tracing useful for research?
A: Yes, for studying agent flows.

80.

Q: Does Runner tracing show model outputs?
A: Yes, if enabled.

Real-World Applications

81.

Q: Example of Runner in chatbots?
A: To control dialogue depth.

82.

Q: Example of Runner in finance bots?
A: To enforce risk constraints.

83.

Q: Example of Runner in customer support?
A: To prevent endless conversations.

84.

Q: Example of Runner in automation?
A: To control workflow execution.

85.

Q: Why is Runner critical in healthcare AI?
A: To enforce safety and stop errors.

86.

Q: Why is Runner critical in trading AI?

A: To avoid infinite risky trades.

87.

Q: Why is Runner important in research assistants?
A: To manage multi-step reasoning.

88.

Q: Which industries rely on Runners?
A: Finance, healthcare, chatbots, automation.

89.

Q: What is a Runner’s role in compliance?
A: To enforce execution policies.

90.

Q: Can Runner be audited?
A: Yes, logs provide audits.

Conceptual Deep Dives

1.

Q: Is Runner deterministic?
A: Execution control is deterministic; agent outputs may vary.

Q: Can Runner change reasoning depth?
A: Yes, by adjusting max_turns.

Q: Can Runner simulate sandboxed execution?
A: Yes, via constraints.

Q: Can Runner balance creativity vs. control?
A: Yes, by managing agent freedom.

Q: Can Runner enforce cost limits?
A: Yes, by limiting API calls.

Q: Does Runner impact latency?
A: Slightly, due to control logic.

Q: Can Runner prioritize tool calls?
A: Yes, via custom policies.

Q: Can Runner cache outputs?
A: Yes, advanced runners allow caching.

Q: Can Runner alter reward signals in RL agents?
A: Yes, by customizing execution feedback.

Q: Final role of Runner in Agentic AI?
A: To ensure safe, controlled, and goal-oriented execution.