# Python / Agentic AI Exam – Tools (Basic)

100 Questions & Answers

---

## Definition & Basics

1.

   Q: What are Tools in Agentic AI?
   A: External functions, APIs, or utilities that Agents can call to perform tasks.

2.

   Q: True or False: Tools are part of the Agent’s core model.
   A: False – Tools are external, separate from the model.

3.

   Q: Why do Agents need Tools?
   A: To extend capabilities beyond pure text generation.

4.

   Q: Which role do Tools play in Agent reasoning?
   A: They provide factual or actionable results.

5.

   Q: Example: A calculator in an Agent is a?
   A: Tool.

6.

   Q: Example: A weather API in an Agent is a?
   A: Tool.

7.

Q: Can an Agent work without Tools?
A: Yes, but with limited functionality.

8.

Q: What is the key difference between an Agent and a Tool?
A: Agent = decision-maker, Tool = executor.

9.

Q: Can Tools be reused across multiple Agents?
A: Yes.

10.

Q: Which framework term is often used for Tools?
A: Functions, APIs, or plugins.

---

# Tool Properties

11.

Q: Which property enables/disable a Tool?
A: is_enabled.

12.

Q: Which property overrides Tool’s display name?
A: name_override.

13.

Q: Which property customizes Tool’s description?
A: description_override.

14.

Q: Which property defines Tool’s error handling?

A: failure_error_function.

15.

Q: Which property controls Tool’s allowed inputs?
A: input_filter.

16.

Q: Which property controls Tool’s allowed outputs?
A: output_type.

17.

Q: Which property sets Tool’s structured response format?
A: structured_output.

18.

Q: Which property ensures Tool only runs in safe mode?
A: Guardrails.

19.

Q: Which property controls whether Tool logs calls?
A: tracing/logging config.

20.

Q: Which property allows custom output transformation?
A: custom_output_extractor.

---

# Tool Execution

21.

Q: Who triggers Tool execution?
A: Agent (via reasoning).

22.

Q: Who actually runs the Tool?
A: Runner.

23.

Q: If a Tool fails, who handles it?
A: failure_error_function + Runner.

24.

Q: Can multiple Tools run in parallel?
A: Yes.

25.

Q: True or False: An Agent always knows which Tool to pick.
A: False – depends on tool_choice config.

26.

Q: Which config lets Agents automatically choose Tools?
A: tool_choice = “auto”.

27.

Q: Which config forces Tool use?
A: tool_choice = “required”.

28.

Q: Which config disables Tool use?
A: tool_choice = “none”.

29.

Q: What happens if all Tools are disabled?
A: Agent relies only on reasoning.

30.

Q: Which property resets Tool selection mid-run?
A: reset_tool_choice.

# Types of Tools

31.

Q: A calculator tool is an example of?
A: Utility Tool.

32.

Q: A web search API is an example of?
A: Knowledge Tool.

33.

Q: A trading API tool is an example of?
A: Domain-Specific Tool.

34.

Q: A file upload Tool is an example of?
A: IO Tool.

35.

Q: An Agent acting as a Tool is called?
A: Agent-as-Tool.

36.

Q: Which type of Tool provides real-time data?
A: Live API Tool.

37.

Q: Which Tool is used for mathematical reasoning?
A: Calculator Tool.

38.

Q: Which Tool is used for debugging code?
A: Code execution Tool.

39.

Q: Which Tool helps translate text?
A: Translation Tool.

40.

Q: Which Tool sends messages to users?
A: Communication Tool.

---

## Tool Use Behaviour

41.

Q: What is tool_use_behavior?
A: Defines how Agent selects and interacts with Tools.

42.

Q: If tool_use_behavior = “aggressive”, what happens?
A: Agent frequently uses Tools.

43.

Q: If tool_use_behavior = “conservative”, what happens?
A: Agent prefers reasoning over Tool use.

44.

Q: If tool_use_behavior = “balanced”, what happens?
A: Agent decides case-by-case.

45.

Q: Which config influences Tool priority?
A: tool_priority setting.

46.

Q: Can tool_use_behavior be customized per Agent?
A: Yes.

47.

Q: True or False: tool_use_behavior is optional.
A: True – defaults exist.

48.

Q: Who decides when a Tool call is “necessary”?
A: The Agent’s reasoning.

49.

Q: Can tool_use_behavior adapt mid-run?
A: Yes, if reset_tool_choice enabled.

50.

Q: Which config ensures certain Tools must always be used?
A: tool_choice = required.

---

# Tool Failure & Errors

51.

Q: What is a Tool failure?
A: When a Tool call does not return expected output.

52.

Q: Who defines error recovery for Tools?
A: failure_error_function.

53.

Q: What happens if a Tool is disabled mid-run?
A: Agent error or fallback.

54.

Q: Which config prevents Tool from breaking entire Agent run?
A: error handling config.

55.

Q: Which config retries Tool calls?
A: retry policy.

56.

Q: Which config logs Tool errors?
A: error_logger.

57.

Q: Can Tool calls be sandboxed?
A: Yes, for safety.

58.

Q: What happens if a Tool gives harmful output?
A: Guardrails block it.

59.

Q: Can Tool calls escalate to humans?
A: Yes, via escalation policy.

60.

Q: Which config hides Tool errors from users?
A: silent_fail.

---

# Tool Integration

61.

Q: How are Tools usually registered with an Agent?
A: By passing them in Agent initialization.

62.

Q: Example code: Agent(tools=[calculator])  What does this mean?
A: Agent has access to calculator Tool.

63.

Q: Can Tools be dynamically added?
A: Yes.

64.

Q: Can Tools be dynamically removed?
A: Yes.

65.

Q: Which config allows name customization for Tools?
A: name_override.

66.

Q: Which config allows description customization?
A: description_override.

67.

Q: Can Tools be shared between Agents?
A: Yes.

68.

Q: Which property prevents accidental Tool calls?
A: is_enabled = False.

69.

Q: Can Tools be cloned across Agents?
A: Yes.

70.

Q: Can one Tool depend on another Tool?
A: Yes (chained tools).

---

## Advanced Tool Config

71.

Q: What is custom_output_extractor in Tools?
A: A function that transforms Tool results before use.

72.

Q: What is structured_output in Tools?
A: Enforces JSON/schema-based output.

73.

Q: Which config enforces strict schema validation?
A: strict_schema.

74.

Q: Which config defines Tool streaming?
A: streaming_enabled.

75.

Q: Which config defines Tool's expected input type?
A: input_type.

76.

Q: Which config validates Tool's input?
A: input_filter.

77.

Q: Which config defines Tool’s allowed output format?
A: output_type.

78.

Q: Which config logs Tool usage frequency?
A: tracing.

79.

Q: Which config enforces rate limiting for Tools?
A: rate_limit.

80.

Q: Which config forces Tool output transformation?
A: custom_output_extractor.

---

## Tool Use Cases

81.

Q: Which Tool would a Finance Agent use?
A: Trading API Tool.

82.

Q: Which Tool would a Weather Agent use?
A: Weather API Tool.

83.

Q: Which Tool would a Travel Agent use?
A: Flight booking Tool.

84.

Q: Which Tool would a Support Agent use?
A: Ticket system Tool.

85.

Q: Which Tool would a Student Agent use?
A: Search/knowledge Tool.

86.

Q: Which Tool would a Doctor Agent use?
A: Medical database Tool.

87.

Q: Which Tool would a Developer Agent use?
A: Code execution Tool.

88.

Q: Which Tool would a Translator Agent use?
A: Translation API Tool.

89.

Q: Which Tool would a Teacher Agent use?
A: Quiz generation Tool.

90.

Q: Which Tool would a Lawyer Agent use?
A: Legal reference Tool.

---

## Summary & Edge Cases

91.

Q: If no Tools are available, can an Agent still function?
A: Yes, but reasoning-only.

92.

Q: If multiple Tools are available, how does Agent pick?
A: Based on reasoning + tool_choice config.

93.

Q: Can Tools return wrong data?
A: Yes, if API faulty.

94.

Q: Who validates Tool results?
A: Agent reasoning + custom validators.

95.

Q: Can an Agent ignore a Tool's result?
A: Yes, if reasoning decides.

96.

Q: What is the main risk with Tools?
A: Incorrect or unsafe outputs.

97.

Q: What is the main benefit of Tools?
A: Extending Agent capabilities.

98.

Q: Which config ensures Tool safety?
A: is_enabled + guardrails.

99.

Q: Which config ensures Tool flexibility?
A: reset_tool_choice.

100.

Q: What is the ultimate purpose of Tools in Agents?
A: To connect reasoning models with actionable, external capabilities.