**TOOLS**

**1. How many main classes of tools does the Agents SDK provide to let agents take actions?**

A. One  
B. Two  
C. Three  
D. Four

**Answer:** C — The documentation states there are three classes of tools: hosted tools, function calling tools, and agents-as-tools.

**2. Which tool category runs on the LLM servers alongside the AI models?**

A. Function tools  
B. Hosted tools  
C. Agents as tools  
D. None of the above

**Answer:** B — Hosted tools run on LLM servers alongside the AI models.

**3. Which of the following is *not* listed as a hosted tool?**

A. WebSearchTool  
B. ImageGenerationTool  
C. DatabaseQueryTool  
D. LocalShellTool

**Answer:** C — The documentation mentions WebSearchTool, FileSearchTool, ComputerTool, CodeInterpreterTool, HostedMCPTool, ImageGenerationTool, and LocalShellTool, but not DatabaseQueryTool.

**4. The FileSearchTool is used to:**

A. Execute code snippets  
B. Search your OpenAI vector-store files  
C. Browse the web  
D. Control local shell commands

**Answer:** B — FileSearchTool allows retrieving information from your OpenAI Vector Stores.

**5. What does the ComputerTool enable an agent to do?**

A. Search the web  
B. Automate computer use tasks  
C. Read files from disk  
D. Call other agents

**Answer:** B — ComputerTool enables automating computer use tasks.

**6. Which tool allows the LLM to execute code in a sandboxed environment?**

A. CodeInterpreterTool  
B. LocalShellTool  
C. WebSearchTool  
D. HostedMCPTool

**Answer:** A — The CodeInterpreterTool lets the LLM execute code in a sandboxed environment.

**7. Function calling tools are defined as:**

A. Built-in hosted tools only  
B. Agents that act like tools  
C. Any Python function made available as a tool  
D. Debugging utilities

**Answer:** C — Function calling tools allow you to use any Python function as a tool.

**8. When using a Python function as a tool, the tool’s name is derived from:**

A. Its return type  
B. The function name—unless overridden  
C. The first line of its docstring  
D. A random string

**Answer:** B — The name of the tool is taken from the Python function name, unless you provide an override.

**9. What is used to automatically generate the JSON schema for function inputs?**

A. Manual JSON typing  
B. Pydantic, inspect, and griffe  
C. OpenAI API  
D. Hand-coded YAML

**Answer:** B — The SDK uses Python’s inspect module for signatures, griffe for docstrings parsing, and Pydantic for schema creation.

**10. To define a custom function tool manually, which of the following is *not* required?**

A. name  
B. description  
C. params\_json\_schema  
D. OpenAI API key

**Answer:** D — A name, description, params\_json\_schema, and an on\_invoke\_tool handler are required; an OpenAI API key is not part of defining a custom FunctionTool.

**11. What is the purpose of "Agents as tools"?**

A. To allow hosting of LLMs as external services  
B. To allow an agent to use another agent as a tool  
C. To replace function tools with hosted tools  
D. To generate images

**Answer:** B — "Agents as tools" enables you to use an agent as a tool, allowing one agent to call another agent.

**12. Which of the following is not a customization feature under Agents as tools?**

A. Custom output extraction  
B. Conditional tool enabling  
C. Automatic agent training  
D. Customizing tool-agents

**Answer:** C — The documentation mentions customizing tool-agents, custom output extraction, and conditional tool enabling—but not automatic agent training.

**13. Which feature ensures that a tool only executes under certain circumstances?**

A. Hosted tools  
B. Function tools  
C. Conditional tool enabling  
D. Custom output extraction

**Answer:** C — Conditional tool enabling allows a tool to be activated only under specific conditions.

**14. What is *custom output extraction* in the context of Agents as tools?**

A. A method to call external APIs automatically  
B. A way to manually parse and use tool outputs  
C. A debugging endpoint  
D. A visualization tool

**Answer:** B — Custom output extraction allows handling and parsing of the output from tools in a customized manner.

**15. The hosted tool HostedMCPTool exposes:**

A. A remote MCP server’s tools to the model  
B. A file browsing capability  
C. Local shell commands  
D. Web-search interface

**Answer:** A — HostedMCPTool exposes a remote MCP server’s tools to the model.

**16. The decorator used to turn a Python function into a tool is:**

A. @tool  
B. @function\_tool  
C. @make\_tool  
D. @python\_tool

**Answer:** B — The decorator provided by the SDK is @function\_tool.

**17. What will be printed when iterating over agent.tools if tools are FunctionTool instances?**

A. Only the tool descriptions  
B. name, description, and the JSON schema  
C. Only the schema  
D. The agent instructions

**Answer:** B — You can print each tool’s name, description, and params\_json\_schema.

**18. Which library is not mentioned as used under the Function Tools section?**

A. inspect  
B. griffe  
C. Pydantic  
D. FastAPI

**Answer:** D — The documentation references inspect, griffe, and Pydantic, but not FastAPI.

**19. What does LocalShellTool allow?**

A. Execute code in a notebook  
B. Execute shell commands on your local machine  
C. Retrieve vector data  
D. Generate images

**Answer:** B — LocalShellTool runs shell commands on your machine.

**20. If a function tool should override its name, which parameter is used?**

A. name\_override  
B. tool\_name  
C. override\_name  
D. function\_name\_override

**Answer:** A — The name\_override parameter lets you override the default tool name.