Multiple Choice Questions: OpenAI Agents SDK

- Agents Component (30)

Q1) According to the sources, what is the core building block in your apps when using the OpenAI Agents SDK?
a) Tools
b) Runners
c) Agents
d) Models
correct option: c

Q2) An agent is described as a large language model (LLM) configured with what two main elements?
a) Instructions and Context
b) Models and Tools
c) Instructions and Tools
d) Guardrails and Handoffs
correct option: c

Q3) Which of the following is listed as one of the most common properties you'll configure for an agent?
a) handoffs
b) model_settings
c) context
d) instructions
correct option: d

Q4) What is another name for the instructions property of an agent?
a) User input
b) Developer message
c) Agent output

d) Tool schema
correct option: b

Q5) The model property of an agent allows you to specify which LLM to use. What optional configuration can be included with the model?
a) tool_choice
b) model_settings
c) output_type
d) name
correct option: b

Q6) What purpose does the tools property serve for an agent?
a) It defines the context for the agent run.
b) It lists external functions the agent can use.
c) It specifies how the agent should format its output.
d) It determines which other agents the agent can delegate to.
correct option: b

Q7) The sources describe context as a dependency-injection tool. What is passed to Runner.run() and then to every agent, tool, and handoff?
a) The agent's instructions
b) An object representing the context
c) A list of configured tools
d) The agent's output_type
correct option: b

Q8) What kind of Python object can you provide as the context for an agent run?
a) Only Pydantic objects
b) Only dataclasses
c) Only basic types like strings or integers
d) Any Python object
correct option: d

Q9) By default, what type of output do agents produce?

a) JSON objects

b) Pydantic objects

c) Plain text (str)

d) Lists

correct option: c

Q10) To make an agent produce a specific type of output other than plain text, which parameter should you use?

a) output_format

b) result_type

c) output_type

d) return_type

correct option: c

Q11) When you pass an output_type parameter to an agent, what does this tell the model to use instead of regular plain text responses?

a) Dynamic instructions

b) Structured outputs

c) Handoffs

d) Lifecycle hooks

correct option: b

Q12) What mechanism allows an agent to delegate to sub-agents?

a) Guardrails

b) Cloning

c) Context

d) Handoffs

correct option: d

Q13) Providing a list of handoffs to an agent enables a powerful pattern for orchestration. What does this pattern allow?

a) Running checks on user input in parallel.

b) Duplicating agents with different properties.

c) Orchestrating modular, specialized agents.
d) Observing the lifecycle of an agent.
correct option: c

Q14) In addition to providing instructions when creating an agent, how else can instructions be provided?
a) Via an environment variable
b) Via a function that returns the prompt
c) Via the context object
d) Via the model_settings
correct option: b

Q15) A function providing dynamic instructions will receive which parameters?
a) Only the context object
b) The agent and model_settings
c) The agent and context
d) Only the user input
correct option: c

Q16) What property allows you to hook into the agent lifecycle to observe events or pre-fetch data?
a) events
b) hooks
c) lifecycle
d) observers
correct option: b

Q17) To hook into the agent lifecycle, you should subclass which class?
a) AgentLifecycle
b) AgentObserver
c) AgentHooks
d) LifecycleEvents
correct option: c

Q18) What feature allows you to run checks or validations on user input in parallel to the agent running?
a) Handoffs
b) Guardrails
c) Context management
d) Tracing
correct option: b

Q19) How can you duplicate an existing Agent object while optionally changing some of its properties?
a) By using the copy() method
b) By serializing and deserializing the agent
c) By creating a new agent instance and manually copying properties
d) By using the clone() method
correct option: d

Q20) When duplicating an agent using clone(), can you change any of its properties during the cloning process?
a) No, clone() creates an exact duplicate only.
b) Yes, but only the name and instructions.
c) Yes, you can optionally change any properties you like.
d) Yes, but only properties related to the model.
correct option: c

Q21) Supplying a list of tools to an agent means the LLM will always use one of them. True or False?
a) True
b) False
correct option: b

Q22) What ModelSettings property can you set to force tool use by the LLM?
a) force_tool
b) tool_selection
c) tool_choice

d) use_tools

correct option: c

Q23) What is the effect of setting ModelSettings.tool_choice to "auto"?

a) It requires the LLM to use a tool, intelligently deciding which one.

b) It requires the LLM *not* to use a tool.

c) It allows the LLM to decide whether or not to use a tool.

d) It forces the LLM to use a specific tool named "auto".

correct option: c

Q24) What is the effect of setting ModelSettings.tool_choice to "required"?

a) It requires the LLM to use a tool, and it can intelligently decide which tool.

b) It allows the LLM to decide whether or not to use a tool.

c) It requires the LLM *not* to use a tool.

d) It forces the LLM to use a specific tool named "required".

correct option: a

Q25) What is the effect of setting ModelSettings.tool_choice to "none"?

a) It requires the LLM to use any available tool.

b) It allows the LLM to decide whether or not to use a tool.

c) It requires the LLM *not* to use a tool.

d) It forces the LLM to use a specific tool named "none".

correct option: c

Q26) If you set ModelSettings.tool_choice to a specific string like "my_tool", what does this require the LLM to do?

a) Use any tool except "my_tool".

b) Use the specific tool named "my_tool".

c) Decide whether or not to use "my_tool".

d) Use any tool *or* generate a text response.

correct option: b

Q27) According to the sources, to prevent infinite loops, what does the framework automatically do with tool_choice after a tool call?
a) Sets it to "required"
b) Sets it to "none"
c) Sets it to "auto"
d) Removes the tool_choice setting entirely
correct option: c

Q28) Why does the framework automatically reset tool_choice to "auto" after a tool call?
a) To allow the user to provide new instructions.
b) To give the LLM a chance to end the conversation.
c) To prevent infinite loops where the LLM repeatedly calls a tool.
d) To enable guardrails to run checks.
correct option: c

Q29) How can the automatic reset behavior of tool_choice after a tool call be configured?
a) Via agent.configure_tool_choice_reset.
b) Via ModelSettings.tool_choice_reset.
c) Via agent.reset_tool_choice.
d) It cannot be configured, it's a fixed behavior.
correct option: c

Q30) If you want the agent to completely stop after a tool call and use the tool output as the final response without further LLM processing, which property should you set on the agent?
a) stop_after_tool_call = True
b) tool_use_behavior = "stop_on_first_tool"
c) tool_choice = "none"
d) end_on_tool_output = True
correct option: b

- Running Agents (30)

Q1) Which class is used to run agents in the OpenAI Agents SDK?
a) Agent
b) Model
c) Tool
d) Runner
Correct option: d

Q2) How many options are listed for running agents via the Runner class?
a) 1
b) 2
c) 3
d) 4
Correct option: c

Q3) Which method of the Runner class is used for synchronous execution?
a) Runner.run()
b) Runner.run_sync()
c) Runner.run_streamed()
d) Runner.async_run()
Correct option: b

Q4) Which method of the Runner class runs asynchronously and returns a RunResult?
a) Runner.run_sync()
b) Runner.run_streamed()
c) Runner.async_run()
d) Runner.run()
Correct option: d

Q5) Which method of the Runner class runs asynchronously, calls the LLM in streaming mode, and returns a RunResultStreaming?
a) Runner.run()

b) Runner.run_sync()
c) Runner.run_streamed()
d) Runner.stream_run()
Correct option: c

Q6) When using Runner.run(), what two main inputs do you provide to start the agent loop?
a) An Agent object and run_config
b) An Agent object and the LLM model
c) A Runner object and the input string
d) A starting agent and input
Correct option: d

Q7) In the agent loop, what happens after the LLM produces its output if it's not a final_output, handoff, or tool call?
a) The loop ends with an error.
b) The framework raises a MaxTurnsExceeded exception.
c) The framework automatically attempts a handoff.
d) This state is not described as possible in the loop.
Correct option: d

Q8) According to the description of the agent loop, what happens if the LLM produces tool calls?
a) The loop ends, returning the tool calls.
b) The loop ends with an error.
c) The tool calls are run, results appended, and the loop re-runs.
d) The framework automatically performs a handoff.
Correct option: c

Q9) The agent loop continues until what condition is met?
a) The LLM produces tool calls.
b) The LLM performs a handoff.
c) The LLM returns a final_output.
d) The user interrupts the process.
Correct option: c

Q10) What happens if the agent run exceeds the max_turns passed to the run methods?
a) The loop ends and the current state is returned.
b) A MaxTurnsExceeded exception is raised.
c) The framework automatically attempts a handoff.
d) The LLM is instructed to generate a final_output.
Correct option: b

Q11) What is considered a "final output" by the agent loop?
a) Any text output from the LLM.
b) Text output with the desired type, regardless of tool calls.
c) Any output that includes tool calls.
d) Text output with the desired type, and no tool calls.
Correct option: d

Q12) What feature allows you to receive events from the LLM as they are received during the run?
a) RunResult
b) RunResultBase
c) RunResultStreaming
d) Streaming
Correct option: d

Q13) The run_config parameter is used to configure what aspect of the agent run?
a) The agent's specific instructions.
b) Global settings for the agent run.
c) The definition of tools for the agent.
d) How handoffs are handled.
Correct option: b

Q14) Which run_config property allows you to set a global LLM model to use, overriding agent-specific models?
a) model_settings
b) model_provider

c) model_name
d) model
Correct option: d

Q15) Which run_config property allows you to override agent-specific model settings like temperature or top_p?
a) model
b) model_provider
c) model_settings
d) temperature_settings
Correct option: c

Q16) Which run_config properties allow you to include guardrails on all runs?
a) handoff_input_filter
b) tracing_disabled, trace_include_sensitive_data
c) workflow_name, trace_id, group_id
d) input_guardrails, output_guardrails
Correct option: d

Q17) What is the purpose of the handoff_input_filter in run_config?
a) To block certain agents from receiving handoffs.
b) To apply a global filter to inputs sent to the new agent during a handoff, if not already specified.
c) To filter the output of the handoff agent.
d) To determine which agent should receive a handoff.
Correct option: b

Q18) Which run_config property allows you to disable tracing for the entire run?
a) trace_metadata
b) tracing_disabled
c) trace_id
d) workflow_name
Correct option: b

Q19) Which run_config property configures whether traces will include potentially sensitive data?
a) tracing_disabled
b) trace_include_sensitive_data
c) trace_metadata
d) trace_sensitive_data
Correct option: b

Q20) What are workflow_name, trace_id, and group_id in run_config related to?
a) Guardrails configuration
b) Model settings overrides
c) Handoff filtering
d) Tracing
Correct option: d

Q21) A single call to any of the Runner run methods represents what in the context of a chat conversation?
a) A single LLM call
b) A single agent execution
c) A single logical turn
d) A single tool call
Correct option: c

Q22) How can you obtain the inputs for the next turn in a conversation after an agent run completes?
a) By manually reconstructing the input list.
b) By using the Runner.next_input() method.
c) By calling RunResultBase.to_input_list().
d) By accessing the inputs property of the RunResult.
Correct option: c

Q23) What is the base class for all exceptions raised in the OpenAI Agents SDK?
a) RunnerException
b) AgentException

c) SDKException
d) AgentsException
Correct option: d

Q24) What exception is raised when the agent run exceeds the configured maximum number of turns?
a) TurnsExceededError
b) MaxTurnsReachedException
c) MaxTurnsExceeded
d) LoopLimitExceeded
Correct option: c

Q25) When is a ModelBehaviorError raised according to the sources?
a) When a guardrail is tripped.
b) When the run exceeds max_turns.
c) When the model produces invalid outputs (e.g., malformed JSON, non-existent tools).
d) When the user makes an error using the SDK.
Correct option: c

Q26) What type of exception is raised when the person writing code using the SDK makes an error?
a) ModelBehaviorError
b) UserError
c) AgentsException
d) SDKUsageError
Correct option: b

Q27) Which exceptions are raised when a guardrail is tripped?
a) AgentsException, UserError
b) MaxTurnsExceeded, ModelBehaviorError
c) InputGuardrailTripwireTriggered, OutputGuardrailTripwireTriggered
d) GuardrailError, TripwireException
Correct option: c

Q28) What input types are supported for the input parameter when using the Runner.run method?
a) Only a string (user message).
b) Only a list of input items.
c) A string (user message) or a list of input items.
d) A dictionary representing the context.
Correct option: c

Q29) The input items provided to Runner.run are described as being items in which API?
a) OpenAI Chat Completions API
b) OpenAI Assistants API
c) OpenAI Embeddings API
d) OpenAI Responses API
Correct option: d

Q30) Where can you find the full list of exceptions raised by the SDK?
a) In the Runner class documentation.
b) In agents.exceptions.
c) In the SDK quickstart guide.
d) In the RunResult object.
Correct option: b

- Results (20)

Q) What object is returned when you call the Runner.run() or Runner.run_sync() methods?
a) RunResultBase
b) RunResultStreaming
c) RunCompletion
d) RunResult
correct option: d

Q) What object is returned when you call the Runner.run_streamed() method?
a) RunResult

b) RunStreamResult
c) RunResultStreaming
d) RunBaseStreaming
correct option: c

Q) Both RunResult and RunResultStreaming inherit from which base class, which contains most of the useful information?
a) RunnerBase
b) ResultBase
c) RunResultBase
d) AgentResultBase
correct option: c

Q) What property of the result object contains the final output of the last agent that ran?
a) output
b) final_output
c) agent_output
d) last_output
correct option: b

Q) If the last agent run had an output_type defined, what is the type of the final_output property?
a) str
b) Any
c) An object of the type last_agent.output_type
d) A dictionary
correct option: c

Q) Why is the final_output property statically typed as Any?
a) Because it can be a string or an object.
b) To indicate that its type is not known until runtime.
c) Because of the possibility of handoffs, where any agent could be the last one.
d) To allow for flexibility in return types.
correct option: c

Q) Which method is used to convert a RunResultBase object into an input list suitable for a subsequent agent run?
a) result.to_list()
b) result.generate_input()
c) result.to_input_list()
d) result.create_input()
correct option: c

Q) What does the result.to_input_list() method concatenate to create the new input list?
a) The original input provided to run and the final_output.
b) The original input provided to run and the new_items generated during the run.
c) The new_items and the final_output.
d) The raw_responses and the final_output.
correct option: b

Q) What information is stored in the last_agent property of the result object?
a) The first agent that started the run.
b) The agent that handled the initial user input.
c) The last agent that completed its run.
d) A list of all agents involved in the run.
correct option: c

Q) Why might storing the last_agent be useful for the next user input?
a) To always restart the conversation with the initial agent.
b) To determine if a handoff occurred.
c) To potentially resume interaction with the specific agent that handled the previous turn, especially after a handoff.
d) To calculate the total runtime of the last agent.
correct option: c

Q) What property contains the items generated *during* the run, wrapped in RunItem objects?
a) generated_items
b) run_items

c) new_items

d) intermediate_items

correct option: c

Q) What does a RunItem object wrap?

a) The configuration for the next agent.

b) The raw item generated by the LLM.

c) The tool call outputs.

d) The guardrail results.

correct option: b

Q) Which RunItem type indicates a message generated directly by the LLM?

a) MessageItem

b) LLMOutputItem

c) MessageOutputItem

d) TextItem

correct option: c

Q) Which RunItem type indicates that the LLM specifically called the handoff tool?

a) HandoffItem

b) CallHandoffItem

c) HandoffCallItem

d) AgentHandoffItem

correct option: c

Q) What does a HandoffOutputItem indicate?

a) That the LLM requested a handoff.

b) That a handoff process actually occurred.

c) That a handoff tool call failed.

d) The target agent for a potential handoff.

correct option: c

Q) From which RunItem type can you access the source and target agents involved in a transfer of control?

a) HandoffCallItem

b) HandoffOutputItem

c) ToolCallItem

d) MessageOutputItem

correct option: b

Q) Which RunItem type indicates that the LLM invoked a tool (before the tool runs)?

a) ToolInvocationItem

b) ToolCallItem

c) AgentToolItem

d) FunctionCallItem

correct option: b

Q) Which RunItem type indicates that a tool was called and provides access to the tool's output?

a) ToolCallItem

b) ToolOutputItem

c) ToolCallOutputItem

d) ToolResponseItem

correct option: c

Q) What properties of the result object contain information if guardrails were used during the run?

a) guardrail_status, guardrail_details

b) input_guardrail_results, output_guardrail_results

c) guardrail_input, guardrail_output

d) tripped_guardrails

correct option: b

Q) Which property contains the ModelResponses generated by the LLM during the run?

a) llm_responses

b) raw_responses

c) model_outputs
d) agent_responses
correct option: b

<u>- **Streaming** </u>(20)

Q) What is the primary benefit of using streaming when running agents?
a) It reduces the overall runtime of the agent.
b) It allows you to subscribe to updates as the agent run proceeds, showing progress and partial responses to the end-user.
c) It significantly lowers the cost of the API calls.
d) It automatically handles guardrail violations.
correct option: b

Q) Which method do you call on a Runner object to initiate a streaming agent run?
a) Runner.run()
b) Runner.run_sync()
c) Runner.run_streamed()
d) Runner.stream()
correct option: c

Q) What object is returned when you call Runner.run_streamed()?
a) RunResult
b) RunBaseStreaming
c) RunCompletion
d) RunResultStreaming
correct option: d

Q) After obtaining a RunResultStreaming object, which method do you call on it to get an async stream of event objects?
a) result.get_events()
b) result.stream_events()
c) result.get_stream()

d) result.events_stream()
correct option: b

Q) What is the base type of the objects yielded by result.stream_events()?
a) StreamData
b) RunEvent
c) AgentEvent
d) StreamEvent
correct option: d

Q) Which type of stream event passes raw events directly from the LLM in OpenAI Responses API format?
a) RawStreamEvent
b) RawResponsesStreamEvent
c) LLMEvent
d) AgentRawEvent
correct option: b

Q) According to the sources, what are RawResponsesStreamEvent events useful for?
a) Detecting tool calls.
b) Informing when an item has been fully generated.
c) Streaming response messages to the user as soon as they are generated, potentially token-by-token.
d) Getting updates when the current agent changes.
correct option: c

Q) In RawResponsesStreamEvent, what properties does each event have according to the source?
a) status and output
b) item and type
c) type and data
d) agent and input
correct option: c

Q) Which RawResponsesStreamEvent type is specifically mentioned as providing text deltas, useful for token-by-token streaming?
a) response.created
b) response.tool_call
c) response.output_text.delta
d) response.done
correct option: c

Q) What type of stream event is described as "higher level" and informs you when an item has been *fully generated*?
a) RawResponsesStreamEvent
b) AgentUpdatedStreamEvent
c) RunItemStreamEvents
d) FullItemStreamEvent
correct option: c

Q) RunItemStreamEvents allow you to push progress updates at which levels?
a) Token-by-token
b) Only tool calls
c) "message generated", "tool ran", etc.
d) Agent changes only
correct option: c

Q) In a RunItemStreamEvents object, what property holds the generated item itself?
a) data
b) item
c) output
d) generated
correct option: b

Q) Which RunItemStreamEvents item type indicates that the LLM invoked a tool?
a) "tool_call_output_item"
b) "tool_item"
c) "tool_call_item"

d) "invocation_item"
correct option: c

Q) Which RunItemStreamEvents item type indicates that a tool was called and provides access to its output?
a) "tool_call_item"
b) "tool_output_item"
c) "tool_call_output_item"
d) "tool_response_item"
correct option: c

Q) Which RunItemStreamEvents item type indicates a message generated by the agent/LLM?
a) "message_item"
b) "output_item"
c) "message_output_item"
d) "text_item"
correct option: c

Q) What type of stream event gives you updates when the current agent changes?
a) AgentStreamEvent
b) AgentChangeStreamEvent
c) AgentUpdatedStreamEvent
d) HandoffStreamEvent
correct option: c

Q) What is a scenario mentioned in the source that would trigger an AgentUpdatedStreamEvent?
a) The LLM generates a message.
b) The LLM calls a tool.
c) As the result of a handoff.
d) When the run completes.
correct option: c

Q) In an AgentUpdatedStreamEvent object, what property would you typically access to get information about the *new* agent?
a) agent
b) updated_agent
c) new_agent
d) current_agent
correct option: c

Q) The example code for RunItemStreamEvents and AgentUpdatedStreamEvent shows filtering out which type of event using continue?
a) agent_updated_stream_event
b) run_item_stream_event
c) raw_response_event
d) tool_call_item
correct option: c

Q) Compared to RawResponsesStreamEvent, RunItemStreamEvents are better suited for:
a) Displaying text output token-by-token.
b) Providing higher-level progress updates like "tool ran" or "message generated".
c) Receiving direct data from the LLM API.
d) Quickly identifying handoffs.
correct option: b

- **Tools (30)**

Q) What is the primary purpose of tools in the OpenAI Agent SDK?
a) To define the agent's persona
b) To fetch data, run code, call external APIs, and use a computer
c) To handle handoffs between agents
d) To configure guardrail policies

correct option: b

Q) How many main classes of tools are there in the Agent SDK?
a) One
b) Two
c) Three
d) Four

correct option: c

Q) Which tool class runs on LLM servers alongside the AI models?
a) Function calling
b) Hosted tools
c) Agents as tools
d) Custom tools

correct option: b

Q) Which tool class allows you to use any Python function as a tool?
a) Hosted tools
b) Agents as tools
c) Function calling
d) Built-in tools

correct option: c

Q) Which tool class allows agents to call other agents without necessarily handing off control?
a) Hosted tools
b) Function calling
c) Agents as tools
d) Orchestration tools

correct option: c

Q) Which of these is listed as a Hosted tool offered by OpenAI?
a) CalculatorTool
b) WebSearchTool

c) DatabaseTool
d) EmailTool

correct option: b

Q) What specific hosted tool allows retrieving information from your OpenAI Vector Stores?
a) VectorSearchTool
b) FileSearchTool
c) RetrieveTool
d) DocumentTool

correct option: b

Q) When creating a Function tool from a Python function using the @function_tool decorator, where is the tool name taken from by default?
a) A name parameter in the decorator
b) The function's docstring
c) The name of the Python function
d) The agent's name

correct option: c

Q) When using the @function_tool decorator, where is the tool description taken from by default?
a) A description parameter in the decorator
b) The docstring of the function
c) The function's name
d) A default generic description

correct option: b

Q) How is the schema for the function tool's inputs automatically created from a Python function decorated with @function_tool?
a) It's manually defined in the decorator.
b) It's inferred from the function's argument names and types.

c) It's extracted from the function's docstring.
d) The user must provide a Pydantic model.

correct option: b

Q) Which Python module is used by the SDK to extract the function signature (including type annotations) for automatic tool creation?
a) pydantic
b) griffe
c) json
d) inspect

correct option: d

Q) Which library is used by the SDK to parse docstrings for tool descriptions and argument descriptions?
a) inspect
b) pydantic
c) griffe
d) typing_extensions

correct option: c

Q) Which library is specifically mentioned as being used for schema creation from function arguments for automatic function tools?
a) inspect
b) griffe
c) pydantic
d) json

correct option: c

Q) What docstring format is explicitly mentioned as being supported for automatic parsing by griffe?
a) Markdown
b) ReST (reStructuredText)

c) Google, Sphinx, and Numpy styles
d) Javadoc style

correct option: c

Q) What parameter can you set when using @function_tool to explicitly specify the docstring style if auto-detection fails?
a) docstring_format
b) docstring_style
c) parse_style
d) docstring_parser

correct option: b

Q) How can you disable the automatic parsing of docstring information when using @function_tool?
a) Set the use_docstring_info parameter to False.
b) Remove the docstring from the function.
c) Set the description parameter in the decorator.
d) Provide an empty string for the docstring.

correct option: a

Q) Which Python types can be used as arguments to functions intended to be Function tools?
a) Only primitive types like str and int.
b) Python primitives, Pydantic models, and TypedDicts.
c) Only Pydantic models.
d) Only types with explicit JSON schema definitions.

correct option: b

Q) Can a Function tool created from a Python function using the @function_tool decorator be asynchronous?
a) No, they must be synchronous.
b) Yes, they can be sync or async.

c) Only if they don't take arguments.
d) Only if they return a string.

correct option: b

Q) If a Python function decorated with @function_tool needs access to the run context (RunContextWrapper), where must this argument be placed in the function signature?
a) As the last argument.
b) As the first argument.
c) It can be placed anywhere as long as it's typed correctly.
d) The context is not accessible in function tools.

correct option: b

Q) What parameter can be used with the @function_tool decorator to provide a name for the tool that is different from the function's original name?
a) tool_name
b) name_override
c) alias
d) function_name

correct option: b

Q) When manually creating a FunctionTool object (not using @function_tool), what key argument specifies the async function that executes when the tool is invoked by the LLM?
a) on_tool_call
b) invoke_function
c) on_invoke_tool
d) run_tool

correct option: c

Q) When manually creating a FunctionTool, what format is the tool's argument schema (params_json_schema) provided in?
a) A Pydantic model object.
b) A Python dictionary matching the expected arguments.

c) A JSON schema dictionary.
d) A string representing the schema.

correct option: c

Q) When manually creating a FunctionTool, the on_invoke_tool function receives the tool's arguments in what format as its args parameter?
a) A Python dictionary.
b) A Pydantic model instance.
c) A JSON string.
d) Individual parameters.

correct option: c

Q) What is a key difference in error handling between a Function tool created with @function_tool and one created manually as a FunctionTool object?
a) Manual tools handle errors automatically; decorated functions require explicit handling.
b) Decorated functions can use failure_error_function; manual tools *must* handle errors inside on_invoke_tool.
c) Only decorated functions can re-raise errors.
d) Manual tools have no error handling capabilities.

correct option: b

Q) When using @function_tool, what is the default behavior for handling tool call errors if the failure_error_function parameter is *not* provided?
a) The error is re-raised.
b) The SDK ignores the error.
c) A default_tool_error_function is run, which tells the LLM an error occurred.
d) The agent run immediately stops.

correct option: c

Q) When using @function_tool, what happens if you explicitly pass None to the failure_error_function parameter?
a) The default error handling function is still used.

b) Any tool call errors will be re-raised for you to handle.
c) The error is logged and ignored.
d) The tool call is retried.

correct option: b

Q) What type of error might be re-raised if you set failure_error_function=None for a @function_tool and the model produced invalid JSON when attempting to call the tool?
a) UserError
b) ModelBehaviorError
c) ToolInvocationError
d) RuntimeError

correct option: b

Q) What is a primary use case mentioned in the source for modeling agents as tools?
a) To reduce the complexity of individual agents.
b) To allow agents to directly modify the run context.
c) To allow a central agent to orchestrate a network of specialized agents.
d) To automatically handle handoffs based on tool outputs.

correct option: c

Q) What is the convenience method provided by the Agent SDK to easily turn an existing agent into a tool?
a) agent.as_tool()
b) agent.to_tool()
c) FunctionTool(agent=agent)
d) Runner.run_as_tool(agent)

correct option: a

Q) If you need to configure an agent-as-tool with advanced options not supported by agent.as_tool() (like max_turns), what is the recommended approach according to the source?
a) Modify the agent object directly before calling as_tool.
b) Use Runner.run directly within the implementation of a custom function tool.

c) There is no way to use advanced configurations with agents as tools.
d) Pass a configuration object to the agent.as_tool method.

correct option: b

- **<u>Handoff</u> (20)**

---

1. What is the primary function of handoffs in the OpenAI Agents SDK?
   a) To split a large task among multiple agents simultaneously.
   b) To manage the logging and tracing of agent actions.
   c) To allow an agent to delegate tasks to another agent, especially useful when agents specialize in distinct areas.
   d) To enable agents to access external data sources.
   Correct option: c

2. How are handoffs typically represented to the Large Language Model (LLM)?
   a) As direct commands within the conversation history.
   b) As internal SDK events not visible to the LLM.
   c) As memory updates for the target agent.
   d) As tools.
   Correct option: d

3. By default, what is the naming convention for a handoff tool to an agent named "Refund Agent"?
   a) delegate_to_RefundAgent
   b) refund_agent_handoff
   c) transfer_to_refund_agent
   d) call_RefundAgent
   Correct option: c

4. Which of the following methods can be used to create a handoff for an agent according to the sources?
   a) Only by listing the target Agent object directly in the handoffs parameter.
   b) Only by using the handoff() function.
   c) Both by listing the target Agent object directly and by using the handoff() function.
   d) Handoffs are created automatically when agents are defined.
   Correct option: c

5. What is the purpose of the handoff() function?
   a) To initiate the handoff process immediately.
   b) To specify the agent to hand off to and allow for optional overrides and input filters.
   c) To define the conversation context for the handoff.
   d) To monitor the success or failure of a handoff.
   Correct option: b

6. Which parameter in the handoff() function is used to specify the target agent?
   a) target_agent
   b) agent
   c) delegate_to
   d) next_agent
   Correct option: b

7. What does the tool_name_override parameter in the handoff() function allow you to do?
   a) Change the target agent dynamically.
   b) Override the default description of the handoff tool.
   c) Override the default tool name (transfer_to_<agent_name>) for the handoff.
   d) Set a custom function to be called instead of the tool.
   Correct option: c

8. What is the purpose of the on_handoff parameter in the handoff() function?
   a) To specify the conditions under which the handoff should occur.
   b) To filter the data passed to the next agent.
   c) To define a callback function executed when the handoff is invoked.
   d) To override the agent's instructions for the duration of the handoff.
   Correct option: c

9. The callback function specified by on_handoff can optionally receive LLM-generated input. How is the type of this input controlled?
   a) Automatically inferred by the SDK.
   b) By setting the output_type parameter.
   c) By using a predefined HandoffInputData object.
   d) By setting the input_type parameter in the handoff() function, often using a BaseModel.
   Correct option: d

10. In the example for handoff inputs, what Pydantic model is used to define the expected input data for the "Escalation agent" handoff?

a) HandoffInputData
b) AgentInput
c) RunContextWrapper
d) EscalationData
Correct option: d

11. By default, what information does the new agent see when a handoff occurs?
a) Only the specific input provided by the LLM (if any).
b) Only the last message from the previous agent.
c) A summary of the conversation history.
d) The entire previous conversation history.
Correct option: d

12. How can you change the information the new agent receives during a handoff?
a) By modifying the target agent's instructions.
b) By setting the on_handoff callback function.
c) By setting an input_filter function that modifies the HandoffInputData.
d) By using the input_type parameter.
Correct option: c

13. What does an input_filter function receive as input?
a) The full RunContextWrapper.
b) The raw text of the conversation history.
c) A HandoffInputData object representing the existing input.
d) The LLM-generated input specified by input_type.
Correct option: c

14. What must an input_filter function return?
a) A boolean indicating whether the handoff should proceed.
b) A string representing the filtered conversation history.
c) None, as it modifies the input in place.
d) A new or modified HandoffInputData object.
Correct option: d

15. The sources mention a common input_filter pattern implemented in agents.extensions.handoff_filters. What is this pattern?
a) Filtering based on specific keywords.
b) Removing sensitive information.
c) Removing all tool calls from the history passed to the next agent.

d) Summarizing the conversation history.
Correct option: c

16. Why is it recommended to include information about handoffs in an agent's prompts?
a) To improve the performance and speed of the handoff.
b) To automatically configure the input_filter.
c) To help LLMs understand and properly utilize the handoffs.
d) To enable the tracing of handoff events.
Correct option: c

17. The sources mention a suggested prefix for agent instructions related to handoffs. Where can this prefix be found?
a) In the handoff() function documentation.
b) In the HandoffInputData class.
c) In agents.extensions.handoff_prompt.RECOMMENDED_PROMPT_PREFIX.
d) It is automatically added by the SDK.
Correct option: c

18. The sources mention a utility to automatically add recommended data to your prompts regarding handoffs. What is this utility called?
a) add_handoff_instructions_to_prompt
b) prompt_with_handoffs
c) RECOMMENDED_PROMPT_PREFIX
d) agents.extensions.handoff_prompt.prompt_with_handoff_instructions
Correct option: d

19. In the basic usage example, how is the billing_agent added to the triage_agent's handoffs list?
a) Using the handoff() function with the agent specified.
b) By listing the billing_agent object directly.
c) Using the on_handoff parameter.
d) It is added automatically because it's another agent.
Correct option: b

20. The on_handoff callback function, if defined, is executed at what point during the handoff process?
a) Before the LLM decides to call the handoff tool.
b) After the target agent has completed its processing.
c) Immediately when the handoff is invoked by the LLM.

d) Before the conversation history is filtered by the input_filter.
Correct option: c

- ## <u>Tracing (10)</u>

---

1. What is the primary purpose of the built-in tracing feature in the OpenAI Agents SDK?
   a) To manage the state and memory of agents across runs.
   b) To define the sequence of agent execution in a workflow.
   c) To provide a comprehensive record of events for debugging, visualization, and monitoring workflows.
   d) To automatically handle the handoff between different agents.
   Correct option: c

2. Which of the following components are the fundamental building blocks of the tracing system as described in the sources?
   a) Agents and Tools
   b) Runners and Pipelines
   c) Traces and Spans
   d) Guardrails and Handoffs
   Correct option: c

3. According to the sources, which of the following events is *not* listed as being wrapped in a specific default span type by the SDK?
   a) LLM generations
   b) Function tool calls
   c) Agent initialization
   d) Handoffs
   Correct option: c

4. Under what specific condition, related to OpenAI API usage, is tracing unavailable according to the sources?
   a) When using custom, non-OpenAI models.
   b) When disabling all guardrails in an agent run.
   c) When operating under a Zero Data Retention (ZDR) policy using OpenAI's APIs.
   d) When running agents asynchronously (run_sync).
   Correct option: c

5. If you want multiple separate calls to Runner.run() to be grouped together within a single trace, what is the recommended method according to the sources?
   a) Manually assign the same trace_id to the RunConfig of each run.
   b) Use the set_trace_processors() function to define a custom grouping logic.
   c) Disable tracing for all individual runs and rely on external logging.
   d) Wrap the entire sequence of Runner.run() calls within a with trace(...) context manager.
   Correct option: d
6. In the context of the Agents SDK tracing, what does a Span represent?
   a) A unique identifier for an entire workflow.
   b) The logical name assigned to a trace.
   c) An operation or unit of work that has a defined start and end time.
   d) A collection of metadata associated with a trace.
   Correct option: c
7. According to the default tracing behavior, which specific type of span is created when an agent performs a handoff to another agent?
   a) agent_span()
   b) function_span()
   c) guardrail_span()
   d) handoff_span()
   Correct option: d
8. How can you prevent the capture of potentially sensitive data within default generation_span() or function_span() entries?
   a) By using the input_filter parameter in handoffs.
   b) By configuring RunConfig.trace_include_sensitive_data to False.
   c) By defining an on_handoff callback function.
   d) Sensitive data capture is not configurable by default.
   Correct option: b
9. Which function allows you to add a new, custom tracing processor to send traces to an *additional* backend, while still maintaining the default behavior of sending traces to the OpenAI backend?
   a) set_trace_processors()
   b) add_trace_processor()
   c) configure_backend_exporter()
   d) replace_default_exporter()
   Correct option: b

10. When tracing is enabled by default and Runner.run() is called, what is the default name given to the overall trace?
    a) "Default Run Trace"
    b) "Workflow Trace"
    c) "Agent trace"
    d) The name is automatically generated based on the agent's name.
    Correct option: c

- **<u>Context Management</u>(10)**

---

1. According to the sources, the OpenAI Agents SDK identifies two main classes of context. What are they?
   a) Global context and User context
   b) Runtime context and Build-time context
   c) Local context and Agent/LLM context
   d) Static context and Dynamic context
   Correct option: c

---

2. In the OpenAI Agents SDK, how is "Local context" primarily managed and accessed within components like tools or lifecycle hooks?
   a) As a global singleton object accessible anywhere.
   b) By passing the context object directly as a standard function argument.
   c) Automatically injected based on function signatures.
   d) Via the RunContextWrapper object, accessed through its context property.
   Correct option: d

---

3. When running an agent using Runner.run(), how is the "Local context" typically provided to the run?
   a) It is automatically detected if defined globally.
   b) By adding it as a key to the input parameter.
   c) By passing it to the context parameter of the run method.
   d) By setting it as an environment variable.
   Correct option: c

4. A critical requirement for "Local context" across a single agent run, according to the sources, is that:
   a) It must be a Pydantic model.
   b) It must be immutable.
   c) It must be serializable.
   d) Every agent, tool, and lifecycle hook in the run must use the same *type* of context.
   Correct option: d

---

5. Which of the following is explicitly listed as a potential use case for "Local context" in the SDK?
   a) Providing the LLM with real-time data updates.
   b) Storing dependencies like a logger object or data fetcher.
   c) Sending sensitive user information directly to the LLM.
   d) Defining the conversation history the LLM sees.
   Correct option: b

---

6. According to the sources, is the object representing "Local context" accessible to the Large Language Model (LLM)?
   a) Yes, it's automatically included in the system prompt.
   b) Yes, it's added to the conversation history by default.
   c) Only if it's a Pydantic model.
   d) No, it is explicitly stated that it is *not* sent to the LLM and is purely a local object.
   Correct option: d

---

7. When an LLM is called within the Agents SDK, what is the *only* data it can see by default, according to the sources?
   a) The "Local context" object.
   b) A summary generated by the SDK.
   c) The conversation history.
   d) Data fetched by retrieval tools.
   Correct option: c

---

8. Which of the following is NOT listed as a method to make *new* data available for the LLM to see by influencing the conversation history?
   a) Adding it to the Agent instructions (system prompt).
   b) Exposing it via function tools for on-demand access.
   c) Passing it directly as the "Local context" object.
   d) Using retrieval or web search tools.
   Correct option: c

---

9. According to the sources, adding context data to the Agent's instructions is particularly useful for:
   a) Data that changes frequently during a run.
   b) Information that is always useful to the LLM (e.g., user's name, current date).
   c) Sensitive data that should not be logged.
   d) Data that the LLM needs to request explicitly.
   Correct option: b

---

10. Making context data available to the LLM via function_tool is described as being useful for:
   a) Providing static, unchanging information.
   b) Data that must be seen at the beginning of every turn.
   c) "On-demand" context that the LLM can decide when it needs and call the tool to fetch.
   d) Summarizing long pieces of text.
   Correct option: c

- **<u>Guardrails</u> (10)**

---

1. What is the primary purpose of guardrails in the OpenAI Agents SDK?
   a) To manage conversational state across turns.
   b) To define the sequence of tool usage.
   c) To provide checks and validations of user input and agent output.
   d) To optimize the underlying model calls.

Correct option: c

---

2. According to the sources, how do guardrails typically run in relation to your agents?
   a) Serially, always completing before the agent starts.
   b) Serially, always completing after the agent finishes.
   c) In parallel to your agents.
   d) They only run if an agent encounters an error.

Correct option: c

---

3. What are the two main types of guardrails described in the source documentation?
   a) Prompt Guardrails and Output Guardrails
   b) System Guardrails and User Guardrails
   c) Input Guardrails and Output Guardrails
   d) Tool Guardrails and Agent Guardrails

Correct option: c

---

4. When do an agent's Input guardrails run, according to the source?
   a) On the final agent output.
   b) Before any tool is called by the agent.
   c) On the initial user input, specifically if the agent is the *first* agent.
   d) On every handoff to another agent.

Correct option: c

---

5. When do an agent's Output guardrails run, according to the source?
   a) On the initial user input.
   b) When the agent receives feedback from a user.
   c) On the final agent output, specifically if the agent is the *last* agent.
   d) Before the agent generates its response text.

Correct option: c

---

6. What is the mechanism called within a guardrail that signals a failure in the input or output check, leading to immediate halting of execution?
   a) A checkpoint

b) A validator
c) A monitor
d) A tripwire

Correct option: d

---

7. According to the sources, what is the immediate consequence when a guardrail's tripwire is triggered?
   a) The guardrail attempts to correct the offending input or output.
   b) The agent logs a warning and continues execution.
   c) The run pauses, awaiting user intervention.
   d) An exception (InputGuardrailTripwireTriggered or OutputGuardrailTripwireTriggered) is raised, and the agent execution halts.

Correct option: d

---

8. Based on the sources, why are guardrails typically defined as properties (e.g., input_guardrails) on the Agent object itself, rather than being passed directly to the Runner.run method?
   a) Because guardrails need direct access to the agent's internal state.
   b) To ensure guardrails are automatically applied to all agents in a sequence.
   c) To simplify configuration by having one central place for all guardrails.
   d) Because guardrails tend to be related to the specific Agent, and collocating the code improves readability.

Correct option: d

---

9. What is the expected return type for a Python function implementing an input or output guardrail in the SDK?
   a) A boolean value indicating success or failure.
   b) A string containing feedback or an error message.
   c) Any arbitrary JSON-serializable object.
   d) A GuardrailFunctionOutput object.

Correct option: d

---

10. The sources suggest a benefit of using a fast/cheap model within a guardrail function, especially when the main agent uses a smart/expensive model. What is this benefit?
    a) The fast model pre-processes the input, making the expensive model faster.
    b) It can quickly detect malicious usage and prevent the expensive model from running, saving time/money.
    c) The fast model provides redundancy if the expensive model fails.
    d) It allows the agent to handle a higher volume of requests concurrently.

Correct option: b

- **Models** **(15)**

---

1. According to the sources, which two flavors of OpenAI models does the SDK come with out-of-the-box support for?
   a) GPT-3.5 and GPT-4
   b) OpenAIResponsesModel and OpenAIChatCompletionsModel
   c) Text Completion models and Chat Completion models
   d) AsyncOpenAI models and SyncOpenAI models

   Correct option: b

---

2. Which of the following OpenAI model flavors is recommended by the SDK sources?
   a) OpenAIChatCompletionsModel
   b) AsyncOpenAI
   c) SyncOpenAI
   d) OpenAIResponsesModel

   Correct option: d

---

3. What is the primary method recommended in the sources for using most non-OpenAI models with the SDK?
   a) Implementing a custom Model interface directly.
   b) Using the set_default_openai_client function.

c) Passing a custom ModelProvider to Runner.run.
d) Utilizing the LiteLLM integration.
Correct option: d

---

4. When using the LiteLLM integration, what prefix should be used for the model name string?
   a) openai/
   b) custom/
   c) provider/
   d) litellm/
   Correct option: d

---

5. Which method for integrating non-OpenAI models is useful when you want to globally use an instance of AsyncOpenAI as the LLM client, typically for providers with an OpenAI compatible API endpoint?
   a) Agent.model setting
   b) Passing a ModelProvider to Runner.run
   c) set_default_openai_client
   d) Using the LiteLLM prefix
   Correct option: c

---

6. Using a ModelProvider at the Runner.run level allows you to:
   a) Specify a model for a single agent instance.
   b) Globally set the default OpenAI client.
   c) Specify a custom model provider for *all* agents in that specific run.
   d) Use the LiteLLM integration without installing the dependency.
   Correct option: c

---

7. Setting the model parameter directly on an Agent instance (Agent.model) enables you to:
   a) Apply a custom model provider to all agents in the run.
   b) Globally set the default model for all agents.
   c) Mix and match different model providers or specific models for different agents within a workflow.

d) Disable tracing for that specific agent.

Correct option: c

---

8. According to the sources, what is a potential reason for mixing and matching different models for different agents within a single workflow?

   a) To use an expensive model for triage and a cheaper model for complex tasks.

   b) To avoid potential conflicts between different model providers.

   c) To use a smaller, faster model for tasks like triage and a larger, more capable model for complex tasks.

   d) To ensure all agents use the same model shape.

   Correct option: c

---

9. Which of the following is one of the ways listed to select a specific model when configuring an Agent?

   a) Passing the name of a model as a string.

   b) Providing a list of potential model names.

   c) Automatically detecting the best model based on instructions.

   d) Using an environment variable specific to the agent.

   Correct option: a

---

10. What is the recommendation regarding model shapes (like OpenAIResponsesModel and OpenAIChatCompletionsModel) within a single workflow, according to the sources?

    a) You should always mix model shapes to leverage different features.

    b) Model shapes are automatically handled by the SDK and don't require user consideration.

    c) It is recommended to use a single model shape for each workflow because the two shapes support a different set of features and tools.

    d) You must ensure all agents use the OpenAIResponsesModel shape.

    Correct option: c

---

11. How can you configure optional model parameters such as temperature for an agent's model?

    a) By passing them directly to the Agent constructor.

    b) By setting them as environment variables.

c) By passing a ModelSettings object to the Agent constructor.
d) Model parameters are not configurable in the SDK.
Correct option: c

---

12. If you are using a non-OpenAI model provider and encounter a Tracing client error 401, what is the likely reason according to the sources?
a) The non-OpenAI provider does not support the Tracing API.
b) The SDK is attempting to use the Responses API with the non-OpenAI provider.
c) You have configured an incorrect API key for the non-OpenAI provider.
d) Traces are uploaded to OpenAI servers, and you do not have an OpenAI API key configured for tracing.
Correct option: d

---

13. What is one recommended way to resolve a Tracing client error 401 when using non-OpenAI models?
a) Install the litellm dependency group.
b) Use the OpenAIChatCompletionsModel instead of OpenAIResponsesModel.
c) Disable tracing entirely using set_tracing_disabled(True).
d) Configure the non-OpenAI provider to upload traces directly.
Correct option: c

---

14. A common issue mentioned when using other LLM providers is that they often do not yet support which OpenAI API shape?
a) The Chat Completions API
b) The Embedding API
c) The Moderation API
d) The Responses API
Correct option: d

---

15. Some model providers have issues with structured outputs, specifically sometimes resulting in a BadRequestError related to response_format.type. What is a described limitation of these providers?
a) They do not support JSON outputs at all.
b) They only support JSON outputs, not text.

c) They support JSON outputs but do not allow you to specify the json_schema to use for the output.
d) They require a separate API call to request structured output.
Correct option: c

- <u>**Configuring the SDK**</u> **(10)**

---

1. According to the sources, what is the primary method the SDK uses by default to find your OpenAI API key upon import?
   a) Reading from a configuration file
   b) Looking for the OPENAI_API_KEY environment variable
   c) Prompting the user for the key
   d) Using a hardcoded default key

Correct option: b

---

2. Which Python function is used to programmatically set the default OpenAI API key if the environment variable method is not suitable?
   a) set_api_key("sk-...")
   b) configure_openai_key("sk-...")
   c) set_default_openai_key("sk-...")
   d) set_openai_api_key("sk-...")

Correct option: c

---

3. To configure the SDK to use a custom instance of AsyncOpenAI (e.g., with a specific base_url), which function should you use?
   a) configure_client(custom_client)
   b) use_openai_client(custom_client)
   c) set_openai_client(custom_client)
   d) set_default_openai_client(custom_client)

Correct option: d

---

4. By default, which specific OpenAI API does the SDK use?
   a) The Chat Completions API
   b) The Embeddings API
   c) The Fine-tuning API
   d) The Responses API

Correct option: d

---

5. Which function allows you to override the default OpenAI API used by the SDK to the Chat Completions API?
   a) set_api_type("chat_completions")
   b) configure_openai_api("chat_completions")
   c) set_default_openai_api("chat_completions")
   d) use_chat_completions_api()

Correct option: c

---

6. What is the default state of tracing in the SDK?
   a) Disabled by default
   b) Enabled, but only logs errors
   c) Enabled, but requires manual activation
   d) Enabled by default

Correct option: d

---

7. Which function is used to disable tracing entirely in the SDK?
   a) disable_tracing(True)
   b) turn_off_tracing()
   c) set_tracing_disabled(True)
   d) stop_tracing()

Correct option: c

---

8. By default, without any custom handler configuration, which logging levels from the SDK's Python loggers are sent to stdout?
   a) DEBUG and INFO

b) INFO and WARNING
c) WARNING and ERROR
d) ERROR only

Correct option: c

---

9. To enable verbose logging, sending more detailed output to stdout, which function is recommended?
a) enable_logging()
b) enable_debug_logging()
c) enable_stdout_logging()
d) enable_verbose_stdout_logging()

Correct option: d

---

10. Which environment variable should be set to 1 to prevent the SDK from logging sensitive LLM inputs and outputs?
a) DISABLE_LLM_LOGS
b) OPENAI_AGENTS_NO_LLM_DATA_LOGS
c) OPENAI_AGENTS_DONT_LOG_TOOL_DATA
d) OPENAI_AGENTS_DONT_LOG_MODEL_DATA

Correct option: d

- **Agent Visualization (10)**

---

1. What is the primary purpose of the Agent Visualization feature in the OpenAI Agents SDK?
a) To monitor agent performance metrics
b) To debug agent code step-by-step
c) To generate a structured graphical representation of agents and their relationships
d) To deploy agent workflows to the cloud

Correct option: c

---

2. Which external tool does the Agent Visualization feature utilize to generate the graphical representation?
   a) Matplotlib
   b) D3.js
   c) Graphviz
   d) Mermaid

Correct option: c

---

3. To install the necessary dependencies for using the Agent Visualization feature, which optional dependency group should be installed via pip?
   a) openai-agents[graph]
   b) openai-agents[render]
   c) openai-agents[visual]
   d) openai-agents[viz]

Correct option: d

---

4. What is the name of the Python function used to generate an agent visualization graph?
   a) generate_graph()
   b) create_visualization()
   c) draw_graph()
   d) visualize_agent()

Correct option: c

---

5. In the generated agent visualization graph, how are Agents visually represented?
   a) Green ellipses
   b) Blue triangles
   c) Red circles
   d) Yellow boxes or rectangles

Correct option: d

---

6. How are Tools visually represented in the agent visualization graph?
   a) Yellow boxes

b) Green ellipses
c) Solid arrows
d) Dotted arrows

Correct option: b

---

7. What graphical element indicates a Handoff from one agent to another in the visualization?
   a) Dotted arrows
   b) Dashed lines
   c) Solid arrows
   d) Double-headed arrows

Correct option: c

---

8. In the generated graph visualization, what do dotted arrows specifically represent?
   a) Agent communication signals
   b) Handoff failures
   c) Data flow between agents
   d) Tool invocations

Correct option: d

---

9. To save the generated graph as a file (e.g., agent_graph.png), what parameter should you provide to the draw_graph function?
   a) output='agent_graph'
   b) save_as='agent_graph.png'
   c) filename='agent_graph'
   d) path='agent_graph.png'

Correct option: c

---

10. By default, draw_graph displays the graph inline. Which method should be called on the result of draw_graph() to display the graph in a separate window?
   a) .display()
   b) .show()

c) .view()
d) .render()

Correct option: c

- **GPT-4.1 Prompting Guide (10)**

---

1. **Compared to GPT-4o, the GPT-4.1 family of models shows significant improvements in which key areas?**
   a) Image generation and audio processing
   b) Data analysis and mathematical calculations
   c) Coding, instruction following, and long context
   d) Conversational fluency and emotional understanding

Correct option: c

---

2. **According to the source, what is a crucial difference in how GPT-4.1 follows instructions compared to its predecessors?**
   a) It infers implicit rules more strongly.
   b) It prioritizes instructions based on user sentiment.
   c) It follows instructions more closely and literally.
   d) It requires less specific prompting.

Correct option: c

---

3. **For optimizing agentic workflows with GPT-4.1, which three key types of reminders are recommended in system prompts?**
   a) Tone, Length, and Creativity
   b) Formatting, Role, and Constraints
   c) Reasoning, Output, and Examples
   d) Persistence, Tool-calling, and Planning (optional)

Correct option: d

---

4. **When integrating tool usage with GPT-4.1 via the OpenAI API, what method is strongly encouraged by the source to minimize errors?**
   a) Manually injecting tool descriptions into the system prompt.
   b) Exclusively using the tools field in the API request.
   c) Providing tool usage examples within the description field.
   d) Relying on the model to infer tool capabilities from context.

Correct option: b

---

5. **In the recommended approach for providing tool usage examples for a complicated tool, where should the examples be placed?**
   a) In the "description" field of the tool parameters.
   b) Immediately after the tool's parameters definition.
   c) In an # Examples section within the system prompt.
   d) As a separate message before the first tool call.

Correct option: c

---

6. **GPT-4.1 is described as not being a "reasoning model." What does this imply about its internal process before answering, by default?**
   a) It performs an extensive internal search for relevant information.
   b) It produces an internal chain of thought.
   c) It always relies solely on provided external context.
   d) It does not produce an internal chain of thought before answering.

Correct option: d

---

7. **To induce explicit, step-by-step planning or "Chain-of-Thought" (CoT) in GPT-4.1, what technique is recommended?**
   a) Providing extensive external context documents.
   b) Setting a higher temperature or top_p value.
   c) Prompting the model to think step by step.
   d) Using an all-caps instruction at the beginning of the prompt.

Correct option: c

---

8. **For long context tasks with GPT-4.1, where is the optimal placement for instructions relative to the provided context?**
   a) Only above the provided context.
   b) Only below the provided context.
   c) At both the beginning and end of the provided context.
   d) Interspersed within the context text.

Correct option: c

---

9. **Which delimiter format is noted as performing particularly poorly when adding a large number of documents or files to the input context for GPT-4.1?**
   a) Markdown
   b) XML
   c) ID: | TITLE: | CONTENT: format
   d) JSON

Correct option: d

---

10. **The GPT-4.1 family features substantially improved capabilities for what coding-related task?**
    a) Writing entirely new codebases
    b) Generating test cases
    c) Debugging code errors
    d) Generating and applying file diffs

Correct option: d
