

```
import os
```

```
from dotenv import load_dotenv
```

```
from swarm_models import OpenAIChat
```

```
from swarms_memory import ChromaDB
```

```
from swarms import Agent
```

```
from swarms.agents.multion_agent import MultiOnAgent
```

```
from swarms.tools.prebuilt.code_interpreter import (
```

```
    SubprocessCodeInterpreter,
```

```
)
```

```
# Load the environment variables
```

```
load_dotenv()
```

```
# Memory
```

```
chroma_db = ChromaDB()
```

```
# MultiOntool
```

```
def multion_tool(
```

```
    task: str,
```

```
    api_key: str = os.environ.get("MULTION_API_KEY"),
```

```
):
```

```
    """
```

Executes a task using the MultiOnAgent.

Args:

task (str): The task to be executed.

api_key (str, optional): The API key for the MultiOnAgent. Defaults to the value of the MULTION_API_KEY environment variable.

Returns:

The result of the task execution.

```
"""
```

```
multion = MultiOnAgent(multion_api_key=api_key)
```

```
return multion(task)
```

```
# Execute the interpreter tool
```

```
def execute_interpreter_tool(
```

```
    code: str,
```

```
):
```

```
    """
```

Executes a single command using the interpreter.

Args:

task (str): The command to be executed.

Returns:

None

```
"""
```

```
out = SubprocessCodeInterpreter(debug_mode=True)
```

```
out = out.run(code)
```

```
return code
```

```
# Get the API key from the environment
```

```
api_key = os.environ.get("OPENAI_API_KEY")
```

```
# Initialize the language model
```

```
llm = OpenAIChat(
```

```
    temperature=0.5,
```

```
    openai_api_key=api_key,
```

```
)
```

```
# Initialize the workflow
```

```
agent = Agent(
```

```
    agent_name="Research Agent",
```

```
    agent_description="An agent that performs research tasks.",
```

```
    system_prompt="Perform a research task.",
```

```
    llm=llm,
```

```
    max_loops=1,
```

```
    dashboard=True,
```

```
    # tools=[multion_tool, execute_interpreter_tool],
```

```
    verbose=True,
```

```
long_term_memory=chroma_db,  
stopping_token="done",  
)  
  
# Run the workflow on a task  
out = agent.run(  
    "Generate a 10,000 word blog on health and wellness, and say done"  
    " when you are done"  
)  
print(out)
```