

```
import datetime
```

```
from typing import List
```

```
from pydantic import BaseModel, Field
```

```
from swarms.tools.base_tool import BaseTool
```

```
from swarms.tools.tool_utils import scrape_tool_func_docs
```

```
time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```
class Thoughts(BaseModel):
```

```
    text: str = Field(..., title="Thoughts")
```

```
    reasoning: str = Field(..., title="Reasoning")
```

```
    plan: str = Field(..., title="Plan")
```

```
class Command(BaseModel):
```

```
    name: str = Field(..., title="Command Name")
```

```
    parameters: dict = Field({}, title="Command Arguments")
```

```
class ResponseFormat(BaseModel):
```

```
    thoughts: Thoughts = Field(..., title="Thoughts")
```

```
    command: Command = Field(..., title="Command")
```

```
response_json = ResponseFormat.model_json_schema()
```

```
tool_usage_browser = """
```

```
```json
{
  "functions": {
    "name": "browser",
    "parameters": {
      "query": "Miami weather"
    }
  }
}
```
```

```
"""
```

```
tool_usage_terminal = """
```

```
```json
{
  "functions": {
    "name": "terminal",
    "parameters": {
```

```
        "code": "uptime"
    }
}
}
'''

''''
```

```
browser_and_terminal_tool = ""
'''
```

```
    "functions": [
        {
            "name": "browser",
            "parameters": {
                "query": "download latest stock data for NASDAQ"
            }
        },
        {
            "name": "terminal",
            "parameters": {
                "cmd": "python analyze_stocks.py"
            }
        }
    ]
}
```

```
...
```

```
"""
```

```
browser_and_terminal_tool_two = """
```

```
...
```

```
{
```

```
  "functions": [
```

```
    {
```

```
      "name": "browser",
```

```
      "parameters": {
```

```
        "query": "download monthly expenditure data"
```

```
      }
```

```
    },
```

```
    {
```

```
      "name": "terminal",
```

```
      "parameters": {
```

```
        "cmd": "python process_expenditures.py"
```

```
      }
```

```
    },
```

```
    {
```

```
      "name": "calculator",
```

```
      "parameters": {
```

```
        "operation": "sum",
```

```
        "numbers": "[output_from_process_expenditures]"
```

```
    }  
    }  
]  
}
```

```
...
```

```
"""
```

```
# Function to parse tools and get their documentation
```

```
def parse_tools(tools: List[BaseTool] = []):
```

```
    tool_docs = []
```

```
    for tool in tools:
```

```
        tool_doc = scrape_tool_func_docs(tool)
```

```
        tool_docs.append(tool_doc)
```

```
    return tool_docs
```

```
# Function to generate the worker prompt
```

```
def tool_usage_worker_prompt(
```

```
    current_time=time, tools: List[callable] = []
```

```
):
```

```
    tool_docs = BaseTool(verbose=True, functions=tools)
```

```
    prompt = f"""
```

**\*\*Date and Time\*\***: {current\_time}

You have been assigned a task that requires the use of various tools to gather information and execute commands.

Follow the instructions provided to complete the task effectively. This SOP is designed to guide you through the structured and effective use of tools.

By adhering to this protocol, you will enhance your productivity and accuracy in task execution.

### ### Constraints

- Only use the tools as specified in the instructions.
- Follow the command format strictly to avoid errors and ensure consistency.
- Only use the tools for the intended purpose as described in the SOP.
- Document your thoughts, reasoning, and plan before executing the command.
- Provide the output in JSON format within markdown code blocks.
- Review the output to ensure it matches the expected outcome.
- Only follow the instructions provided in the SOP and do not deviate from the specified tasks

unless tool usage is not required.

### ### Performance Evaluation

- **\*\*Efficiency\*\***: Use tools to complete tasks with minimal steps.
- **\*\*Accuracy\*\***: Ensure that commands are executed correctly to achieve the desired outcome.
- **\*\*Adaptability\*\***: Be ready to adjust the use of tools based on task requirements and feedback.

### ### Tool Commands

#### 1. **\*\*Browser\*\***

- **\*\*Purpose\*\***: To retrieve information from the internet.

- **Usage**:
- ``{"name": "browser", "parameters": {"query": "search query here"}}`
- Example: Fetch current weather in London.
- Command: ``{"name": "browser", "parameters": {"query": "London weather"}}`

## 2. **Terminal**

- **Purpose**: To execute system commands.
- **Usage**:
- ``{"name": "terminal", "parameters": {"cmd": "system command here"}}`
- Example: Check disk usage on a server.
- Command: ``{"name": "terminal", "parameters": {"cmd": "df -h"}}`

## 3. **Custom Tool** (if applicable)

- **Purpose**: Describe specific functionality.
- **Usage**:
- ``{"name": "custom_tool", "parameters": {"parameter": "value"}}`
- Example: Custom analytics tool.
- Command: ``{"name": "custom_tool", "parameters": {"data": "analyze this data"}}`

## ### Usage Examples

- **Example 1**: Retrieving Weather Information

```
```json
```

```
{tool_usage_browser}
```

```
```
```

- **Example 2**: System Check via Terminal

```
```json  
  
{tool_usage_terminal}  
  
```
```

- **Example 3**: Combined Browser and Terminal Usage

```
```json  
  
{browser_and_terminal_tool}  
  
```
```

- **Example 4**: Combined Browser, Terminal, and Calculator Usage

```
```json  
  
{browser_and_terminal_tool_two}  
  
```
```

### Next Steps

- Determine the appropriate tool for the task at hand.
- Format your command according to the examples provided.
- Execute the command and evaluate the results based on the expected outcome.
- Document any issues or challenges faced during the tool usage.
- Always output the results in the specified format: JSON in markdown code blocks.

### Tools Available



{tool\_docs}

"""

return prompt