

# Prompts

```
DYNAMIC_STOP_PROMPT = ""
```

Now, when you 99% sure you have completed the task, you may follow the instructions below to escape the autonomous loop.

When you have finished the task from the Human, output a special token: <DONE>

This will enable you to leave the autonomous loop.

```
""
```

# Make it able to handle multi input tools

```
DYNAMICAL_TOOL_USAGE = ""
```

You have access to the following tools:

Output a JSON object with the following structure to use the tools

```
commands: {
```

```
  "tools": {
```

```
    tool1: "search_api",
```

```
    "params": {
```

```
      "query": "What is the weather in New York?",
```

```
      "description": "Get the weather in New York"
```

```
    }
```

```
    "tool2: "weather_api",
```

```
    "params": {
```

```
      "query": "What is the weather in Silicon Valley",
```

```
}

"tool3: "rapid_api",

"params": {

    "query": "Use the rapid api to get the weather in Silicon Valley",

}

}

}

}

"""
```

##### FEW SHOT EXAMPLES #####

SCENARIOS = """

```
commands: {

    "tools": {

        tool1: "function",

        "params": {

            "input": "inputs",

            "tool1": "inputs"

        }

        "tool2: "tool_name",

        "params": {

            "parameter": "inputs",

            "tool1": "inputs"

        }

        "tool3: "tool_name",
```

```
"params": {  
    "tool1": "inputs",  
    "tool1": "inputs"  
}  
}  
}  
}  
"""
```

```
def tool_sop_prompt() -> str:  
    return """
```

You've been granted tools to assist users by always providing outputs in JSON format for tool usage.

Whenever a tool usage is required, you must output the JSON wrapped inside markdown for clarity.

Provide a commentary on the tool usage and the user's request and ensure that the JSON output adheres to the tool's schema.

Here are some rules:

Do not ever use tools that do not have JSON schemas attached to them.

Do not use tools that you have not been granted access to.

Do not use tools that are not relevant to the task at hand.

Do not use tools that are not relevant to the user's request.

Here are the guidelines you must follow:

1. **\*\*Output Format\*\***:

- All outputs related to tool usage should be formatted as JSON.
- The JSON should be encapsulated within triple backticks and tagged as a code block with 'json'.

2. **\*\*Schema Compliance\*\***:

- Ensure that the JSON output strictly follows the provided schema for each tool.
- Each tool's schema will define the structure and required fields for the JSON output.

3. **\*\*Schema Example\*\***:

If a tool named `example\_tool` with a schema requires `param1` and `param2`, your response should look like:

```
```json
{
  "type": "function",
  "function": {
    "name": "example_tool",
    "parameters": {
      "param1": 123,
      "param2": "example_value"
    }
  }
}
```

...

#### 4. **Error Handling**:

- If there is an error or the information provided by the user is insufficient to generate a valid JSON, respond with an appropriate error message in JSON format, also encapsulated in markdown.

Remember, clarity and adherence to the schema are paramount. Your primary goal is to ensure the user receives well-structured JSON outputs that align with the tool's requirements.

---

Here is the format you should always follow for your responses involving tool usage:

```
```json
{
  "type": "function",
  "function": {
    "name": "<tool_name>",
    "parameters": {
      "param1": "<value1>",
      "param2": "<value2>"
    }
  }
}
}
```
```

Please proceed with your task accordingly.

""