

Here's a concise cheat sheet for common **Ollama** commands, assuming you're using Ollama to run and manage local large language models. These commands are executed in a terminal or command-line interface where Ollama is installed.


Basic Ollama Commands

Command	Description
<code>ollama --version</code>	Check the installed version of Ollama.
<code>ollama help</code>	Display help information for Ollama commands.
<code>ollama list</code>	List all available models installed locally.
<code>ollama pull <model_name></code>	Download a model from the Ollama registry (e.g., <code>ollama pull llama3</code>).
<code>ollama run <model_name></code>	Run a model interactively (e.g., <code>ollama run llama3</code>).
<code>ollama ps</code>	Show currently running models and their processes.
<code>ollama stop <model_name></code>	Stop a running model.
<code>ollama rm <model_name></code>	Remove a model from your local system.
<code>ollama cp <source_model> <new_model></code>	Copy a model to create a new one with a different name.

Model Management


Command	Description
<code>ollama create <model_name> -f <file></code>	Create a custom model from a Modelfile (e.g., <code>ollama create mymodel -f Modelfile</code>).
<code>ollama show <model_name></code>	Display details about a specific model (e.g., configuration, parameters).
<code>ollama push <model_name></code>	Push a local model to the Ollama registry (if supported).

Running Models

Command	Description	
<code>ollama run <model_name> "<prompt>"</code>	Run a model with a specific prompt non-interactively (e.g., <code>ollama run llama3 "Hello!"</code>).	
<code>ollama run <model_name> --verbose</code>	Run a model with verbose output for debugging.	
<code>ollama run <model_name> --port <port></code>	Specify a custom port for the model server (default is 11434).	

API Interaction

Ollama provides a REST API for programmatic access. Common endpoints:

Command	Description	
<code>curl http://localhost:11434/api/tags</code>	List available models via the API.	
<code>curl -X POST http://localhost:11434/api/generate -d '{"model": "<model_name>", "prompt": "<prompt>"}'</code>	Generate a response from a model via API.	
<code>curl http://localhost:11434/api/ps</code>	Check running models via API.	

Configuration and Setup

- **Default Port:** Ollama runs on `http://localhost:11434` by default.
- **Environment Variables:**
 - `OLLAMA_HOST` : Change the default host/port (e.g., `export OLLAMA_HOST=0.0.0.0:8080`).
 - `OLLAMA_MODELS` : Set the directory for storing models (e.g., `export OLLAMA_MODELS=/path/to/models`).
- **Modelfile:** A file to define custom models (e.g., specify base model, parameters, or system prompts).

Examples

1 Pull and Run a Model:

```
bash
ollama pull llama3
ollama run llama3 "What is the capital of France?"
```

2 Create a Custom Model:

```
bash
echo "FROM llama3" > Modelfile
echo "SYSTEM You are a helpful AI assistant." >> Modelfile
ollama create mycustommodel -f Modelfile
```

3 API Request:

```
bash
curl -X POST http://localhost:11434/api/generate -d '{"model": "llama3"'
```

Notes

- **Model Names:** Use specific model tags (e.g., `llama3:8b`, `mistral:latest`) to specify versions or sizes.
- **System Requirements:** Ensure sufficient CPU/GPU and memory for the model you're running.
- **Documentation:** For advanced usage, check the official Ollama documentation at <https://ollama.com> or GitHub.

If you need more details or specific examples, let me know!