

# Understanding DeepSeek Local Execution Flow

---

## DeepSeek Local Execution: Step-by-Step Guide

### 1 Introduction

DeepSeek LLM allows **local AI processing**, eliminating the need for cloud-based interactions. In this lesson, we focus on:

- ✓ Sending a **prompt** to DeepSeek via the **command line**.
  - ✓ Using Python's **subprocess module** to execute commands.
  - ✓ Handling **successful responses and errors**.
- 

### 2 Execution Flow of an AI Query

To interact with DeepSeek LLM, we follow a **structured execution process**:

- 1 **Set the Prompt** – Define a **query** to send to the AI.
  - 2 **Construct the Command** – Use `subprocess.run()` to execute DeepSeek via CLI.
  - 3 **Execute the Command** – Run DeepSeek and **capture the response**.
  - 4 **Handle Errors** – Check if **execution fails** due to missing dependencies or incorrect parameters.
  - 5 **Process the AI Response** – If successful, **display the output**; if not, show an **error message**.
- 

### 3 Example: Running DeepSeek Locally

To interact with DeepSeek using Python, we run the following command:

```
import subprocess

def run_ollama_prompt(prompt):
    command = f"ollama run deepseek r1 \"{prompt}\""
```

```

try:
    result = subprocess.run(command, shell=True, capture_output=True, t
ext=True)
    if result.returncode == 0:
        print("AI Response:", result.stdout.strip())
    else:
        print("Error:", result.stderr.strip())
except Exception as e:
    print("Execution Failed:", str(e))

# Example prompt
run_ollama_prompt("What is the capital of France?")

```

## 4 Breakdown of the Code

### ✓ Command Construction:

- The **CLI command** to query DeepSeek is:

```
ollama run deepseek r1 "What is the capital of France?"
```

- In Python, we **build this dynamically** using `subprocess.run()`.

### ✓ Subprocess Execution:

- `subprocess.run(command, shell=True, capture_output=True, text=True)`
  - Executes the command in the terminal.
  - `capture_output=True` stores AI responses.

### ✓ Error Handling:

- If the **command runs successfully**, we display the **AI response**.
- If it **fails** (e.g., incorrect DeepSeek installation), we show the **error message**.

## 5 Handling Errors in DeepSeek Execution

If the subprocess fails, common errors include:

**1 Invalid Command:** DeepSeek may not be installed correctly.

**2 Missing Dependencies:** Ensure Python and `ollama` are properly set up.

**3 Execution Errors:** The AI model might be down or misconfigured.

**✓ Solution:**

- Verify the **CLI command works manually** before using Python.
- Check the **installation** with:

```
ollama list
```

- Restart the **local server** if needed.
- 

## **6 Key Takeaways**

- ✓ Learned how to send AI prompts via the CLI and Python.**
- ✓ Understood error handling mechanisms for AI interactions.**
- ✓ Explored how DeepSeek processes requests locally.**