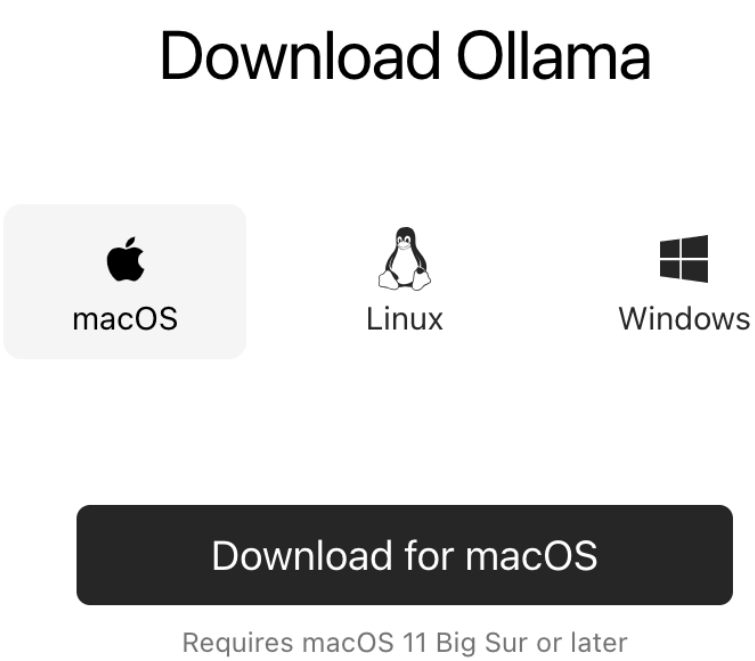
Mit Ollama Sprachmodelle lokal nutzen

Hier auf Pollux Labs konntest du bereits darüber lesen, wie du z.B. die API von OpenAI nutzt, um mit ChatGPT zu interagieren. Aber das geht auch **lokal auf deinem eigenen Rechner** – zwar nicht mit ChatGPT, dafür jedoch mit anderen Sprachmodellen wie Mistral, Gemma, Llama2 und vielen anderen. **Hierfür nutzt du das Tool Ollama.** In diesem Tutorial erfährst du, wie du es **installierst, einrichtest und mit Python mit dem Modell deiner Wahl interagierst.**

Ollama installieren

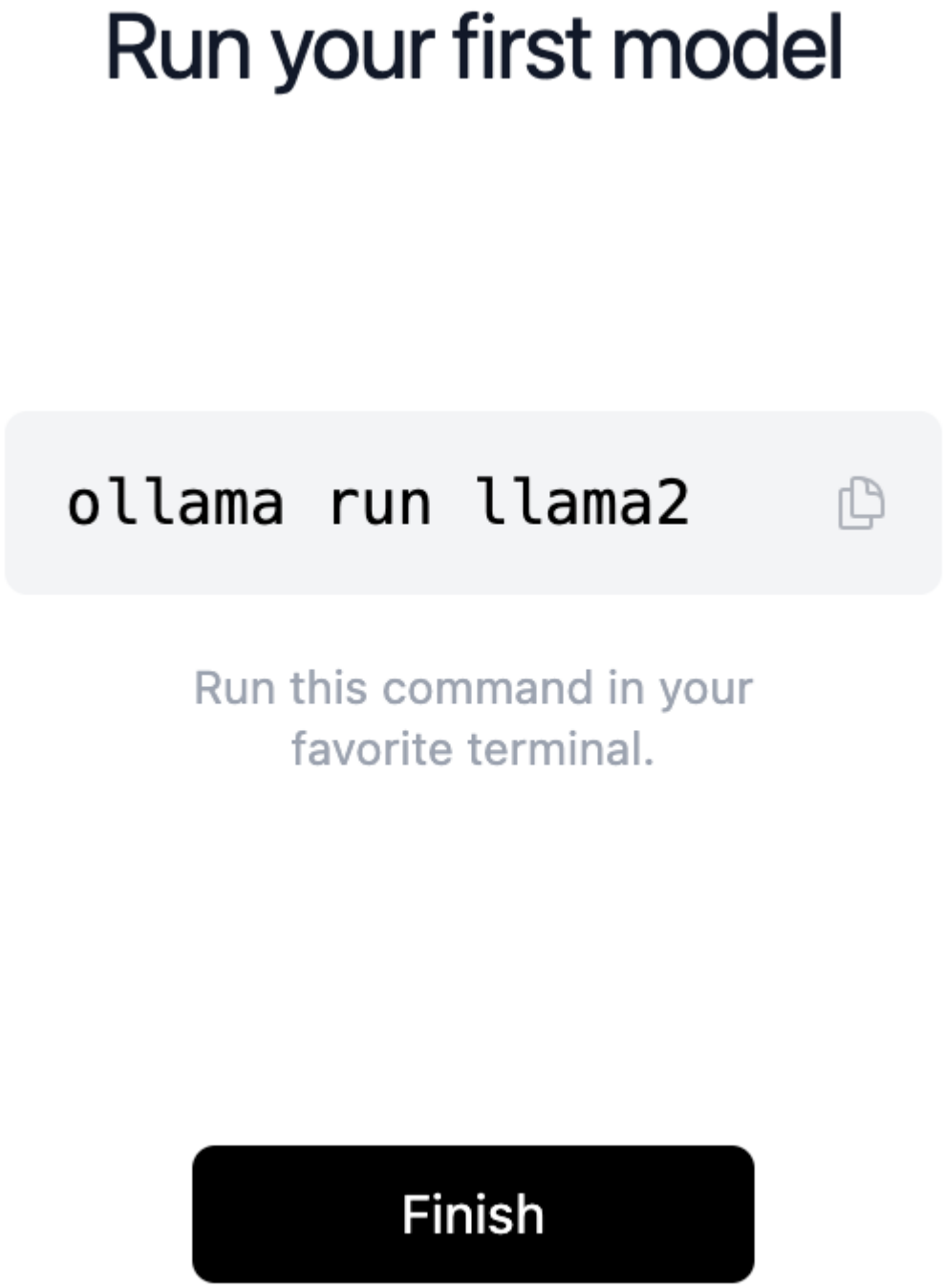
Wenn du einen Mac oder Windows benutzt, musst du Ollama erst hier auf der offziellen Webseite herunterladen.



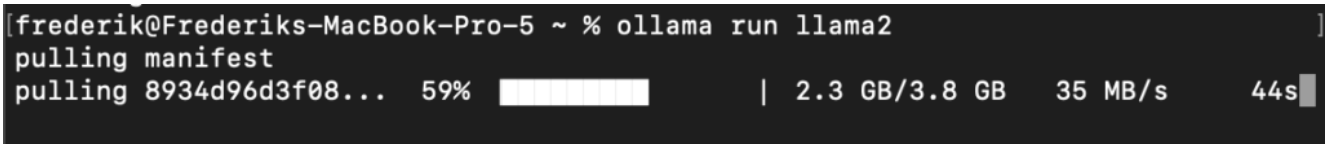
Falls du Linux verwendest, gib im Terminal den folgenden Befehl ein:

```
curl -fsSL https://ollama.com/install.sh | sh
```

Entpacke nach dem Download die ZIP-Datei (Mac) und starte das Programm oder starte direkt die .exe (Windows). Anschließend führt dich ein Wizard durch die nächsten Schritte, damit du Ollama im Terminal verwenden kannst. Am Ende erhältst du den Befehl für einen ersten Test.

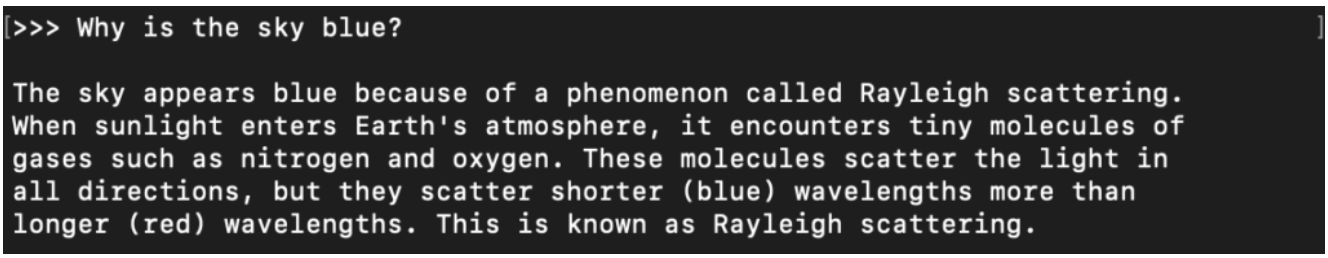


Kopiere den Befehl und gib in im Terminal bzw. der Konsole ein. Anschließend wird das Sprachmodell Llama2 heruntergeladen. Dieses Modell stammt vom Facebook-Konzert Meta. Wie du auf dem Bild unten siehst, ist das mit 3,8 Gigabyte nicht gerade klein – achte also auf genügend Speicherplatz.



Ollama im Terminal verwenden

Um mit deinem ersten Sprachmodell (Llama2) loszulegen, kannst du direkt im selben Fenster bleiben. Du erhältst nach der erfolgreichen Installation eine Eingabeaufforderung, über die du deine erste Frage stellen kannst – so wie du es vermutlich bereits von ChatGPT kennst. Nach wenigen Sekunden erhältst du dann die Antwort ebenfalls im Terminal:



Das funktioniert also schon einmal ganz gut. Das Beispiel oben ist auf Englisch – **du kannst deine Fragen jedoch auch ebenso auf Deutsch stellen. Die Antwort erhältst du von Llama2 jedoch wiederum auf Englisch.** Um Antworten auf Deutsch zu erhalten füge deinem Prompt noch eine entsprechende Anweisung hinzu.

Wenn du deine Session beenden möchtest, gib einfach den Befehl **/bye** ein.

Ein anderes Sprachmodell in Ollama installieren

Du bist natürlich nicht auf Llama2 beschränkt. Auf der Ollama-Webseite sowie auf deren GitHub-Seite kannst du alle verfügbaren Sprachmodelle einsehen. Auf letzterer erfährst du

auch, wieviel Arbeitsspeicher du für die verschiedenen Modelle haben solltest. Versuche es doch als nächstes einmal mit Mistral, einem frei verfügbaren französischen Modell (das auch Deutsch kann). Gib hierfür im Terminal folgenden Befehl ein, nachdem du deine aktive Session mit /bye beendet hast:

```
ollama run mistral
```

Nach der Installation kannst du mit Mistral interagieren, so wie du es vorher mit Llama2 getan hast.

Das Sprachmodell von Mistral ist mit 4,1 GB sogar noch etwas größer als Llama2. Es ist also hilfreich zu wissen, wie du installierte Modelle wieder loswirst. Ganz einfach – Um z.B. Llama2 zu entfernen, gib im Terminal den folgenden Befehl ein:

```
ollama rm llama2
```

Falls du vergessen hast, welche Modelle du gerade installiert hast, hilft dir folgender Befehl weiter:

```
ollama list
```

Ollama mit Python verwenden

Bis jetzt hast du „nur“ im Terminal mit deinem lokalen Sprachmodell kommuniziert. Du kannst aber hierfür natürlich auch ein Python-Script verwenden, ähnlich wie ich es hier schon einmal für ChatGPT beschrieben habe.

Zunächst musst du hierfür die entsprechende Bibliothek installieren:

```
pip install ollama
```

Erstelle nach der erfolgreichen Installation ein leeres Python-Script mit folgendem Inhalt:

```
import ollama
response = ollama.chat(model='mistral', messages=[
  {
    'role': 'user',
    'content': 'Welche Farben können Bären haben? Antworte auf Deutsch.',
  },
])
print(response['message']['content'])
```

Im obigen Script ist wieder das Sprachmodell von Mistral hinterlegt. Falls du ein anderes verwendest, trage es in der zweiten Zeile hinter **model=** ein.

Speiche die Datei nun ab und führe sie aus. Vermutlich wirst du ziemlich lange warten müssen, bis die Antwort erscheint. Das kannst du mit einem Stream verbessern – hierdurch erscheint die lange Antwort Wort für Wort zum Mitlesen. Verwende hierfür den folgenden angepassten Code:

```
import ollama

stream = ollama.chat(
    model='mistral',
     messages=[{'role': 'user', 'content': 'Welche Farben können Bären haben? Anworte auf Deutsch'}],
    stream=True,
)

for chunk in stream:
  print(chunk['message']['content'], end='', flush=True)
```

Weitere Rollen verwenden

Ähnlich wie bei ChatGPT kannst du auch in Ollama in deinen Rollen zuteilen. Wenn du also deine Antworten z.B. immer auf Deutsch erhalten möchtest, hilft dir die Rolle **system** weiter. Dort kannst du die entsprechende Anweisung hinterlegen, sodass die Rolle **user** nur deine Frage enthält:

```
messages=[{'role': 'user', 'content': 'Welche Farben können Bären haben?'},
            {'role': 'system', 'content': 'Antworte auf Deutsch'}],
```

Jetzt kennst du die Grundlagen, um mit Ollama auf deinem eigenen Rechner Sprachmodelle auszuführen und in deine Projekte einzubinden. Es gibt natürlich noch viel mehr zu entdecken: Die verschiedenen Modelle besitzen alle unterschiedliche Fähigkeiten – hier lohnt sich ein intensiver Blick, besonders da die Entwicklung natürlich nicht stehen bleibt.