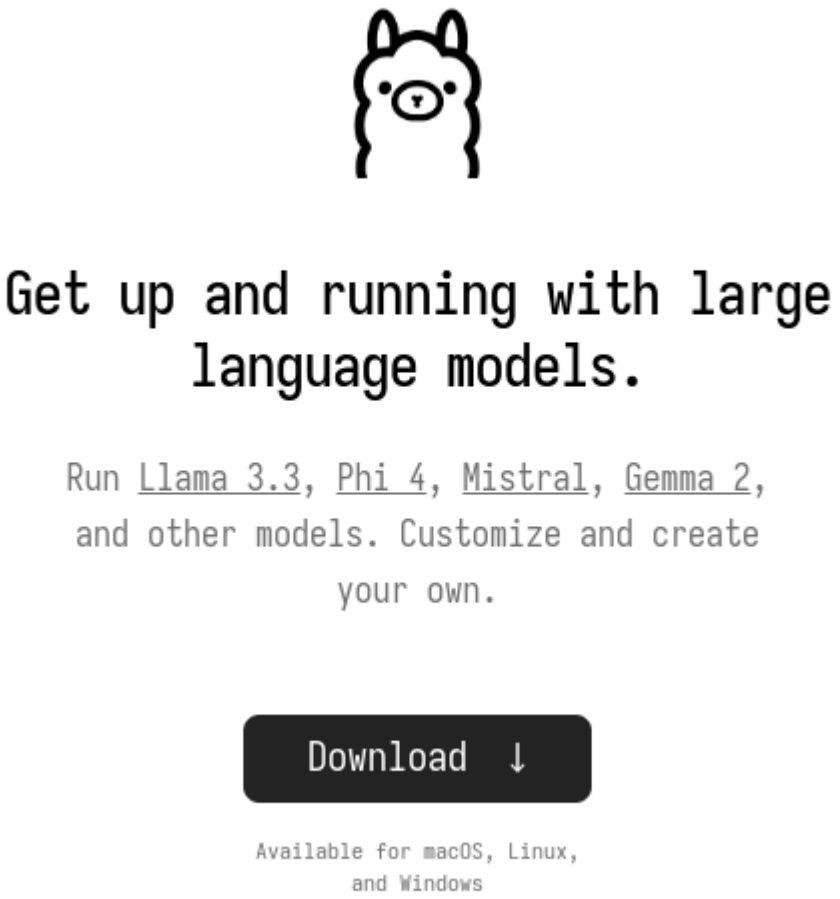
# Hands on 2

## Loading llm locally (40 min)

Local LLMs - Ollama




Ollama is an open-source framework that enables running large language models (LLMs) locally on your personal computer, offering a powerful alternative to cloud-based AI solutions

You can essentially run an llm on your local device for free, without any limitations. The only cap would be the limitations of your system’s CPU and GPU capabilities.

## Installation and Setup

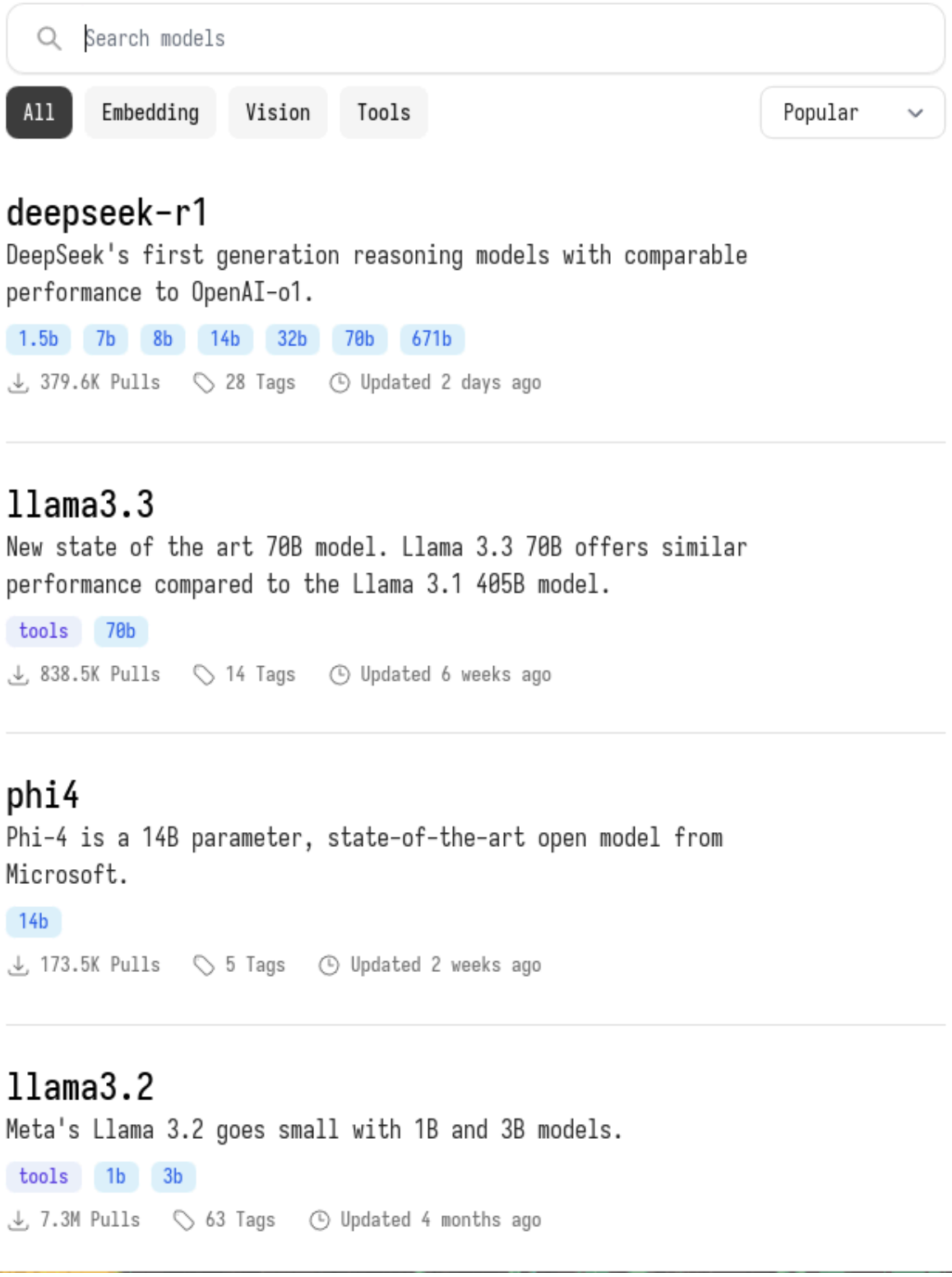
First, download and install Ollama from [ollama.com/download](ollama.com/download)

Verify the download
ollama --version

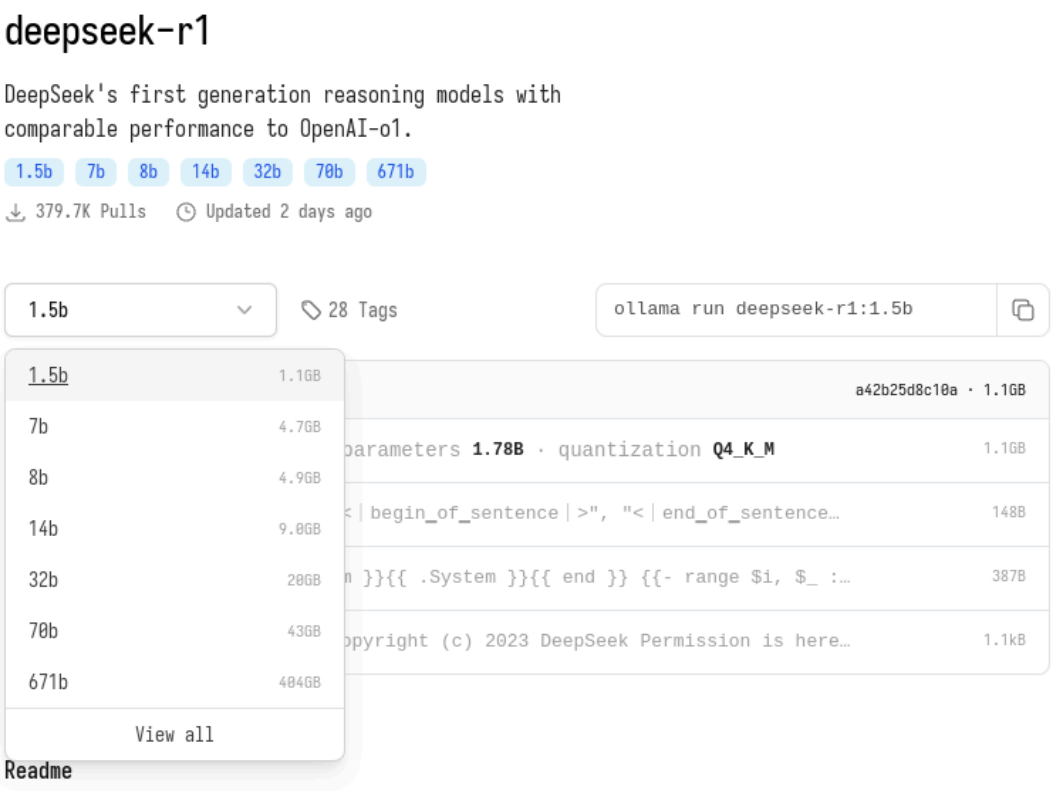
Here's a reference of essential Ollama commands

```
# List available models
ollama list

# Pull a specific model
ollama pull gemma:2b

# Run a model
ollama run gemma:2b

# Remove a model
ollama rm gemma:2b

# Show model details
ollama show gemma:2b
```

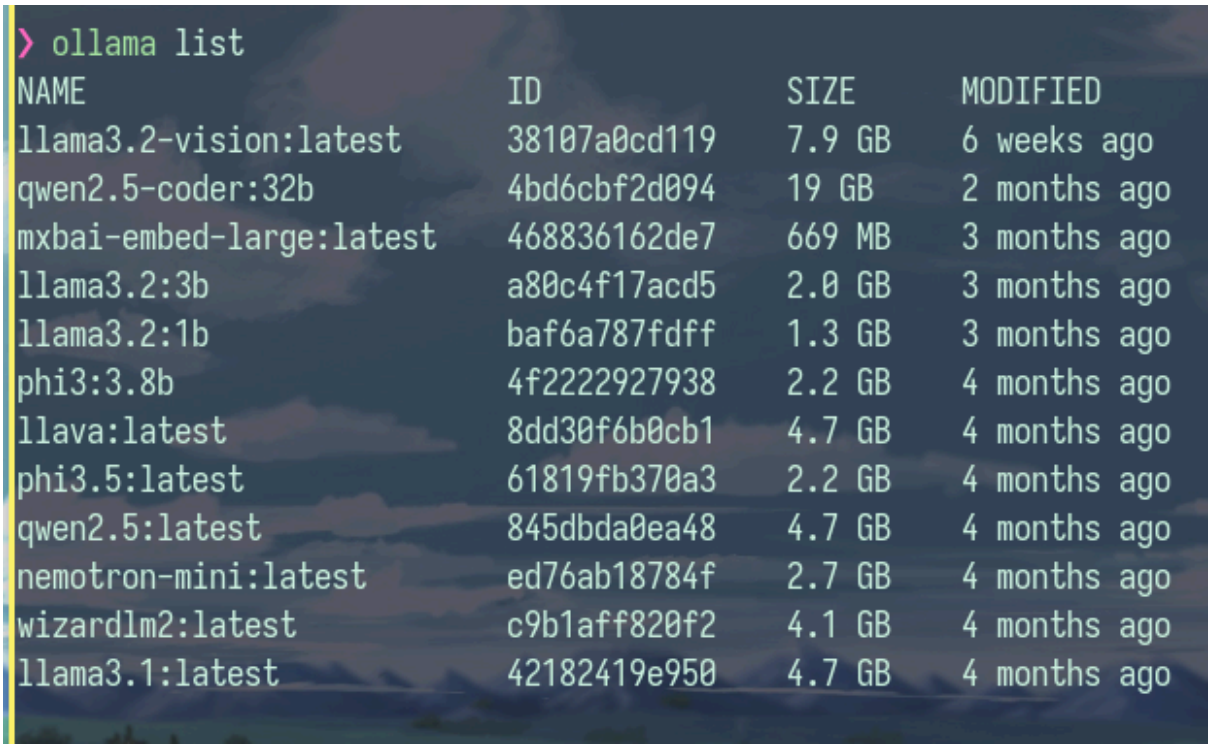
Latest models:



You can select different parameters for your model, the lesser the parameters, the smaller the size of download.

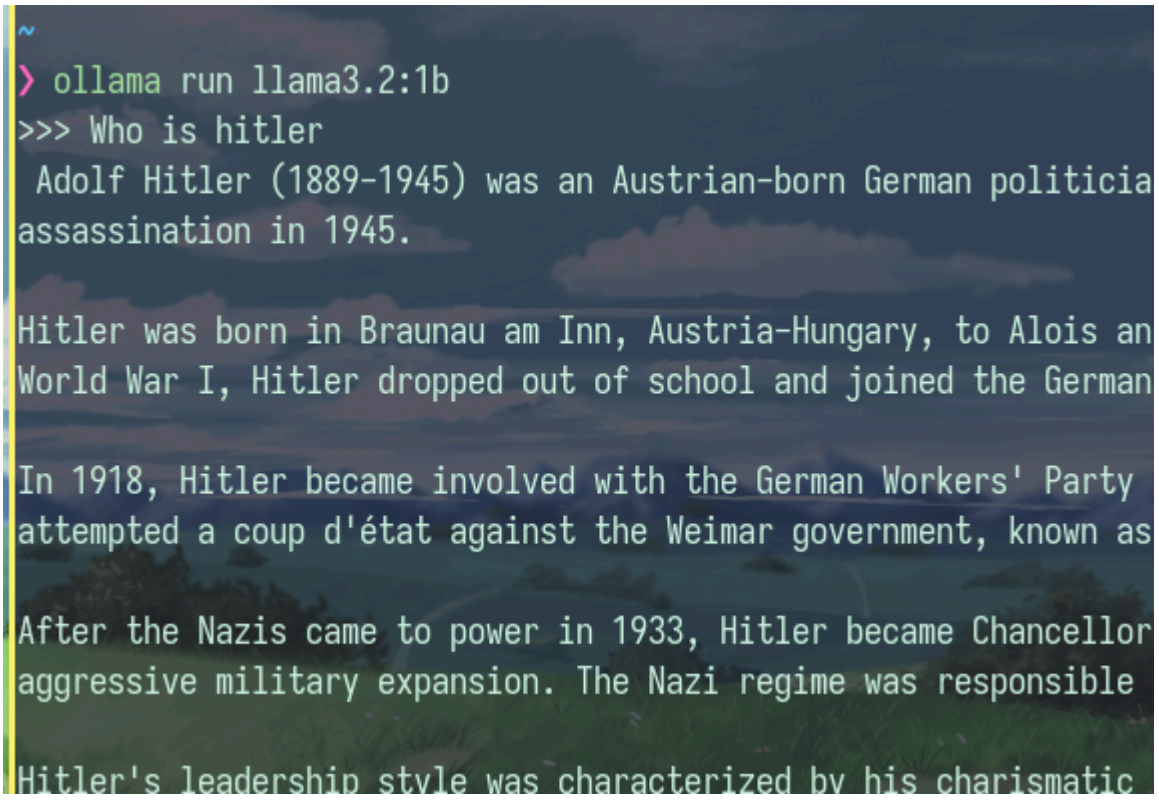


Once downloaded,
In your terminal:
ollama list



```
❯ ollama list
NAME                            ID              SIZE      MODIFIED
llama3.2-vision:latest          38107a0cd119    7.9 GB    6 weeks ago
qwen2.5-coder:32b               4bd6cbf2d094    19 GB     2 months ago
mxbai-embed-large:latest        468836162de7    669 MB    3 months ago
llama3.2:3b                     a80c4f17acd5    2.0 GB    3 months ago
llama3.2:1b                     baf6a787fdff    1.3 GB    3 months ago
phi3:3.8b                       4f2222927938    2.2 GB    4 months ago
llava:latest                    8dd30f6b0cb1    4.7 GB    4 months ago
phi3.5:latest                   61819fb370a3    2.2 GB    4 months ago
qwen2.5:latest                  845dbda0ea48    4.7 GB    4 months ago
nemotron-mini:latest            ed76ab18784f    2.7 GB    4 months ago
wizardlm2:latest                c9b1aff820f2    4.1 GB    4 months ago
llama3.1:latest                 42182419e950    4.7 GB    4 months ago
```

Ollama run <model_name>



```
~
❯ ollama run llama3.2:1b
>>> Who is hitler
 Adolf Hitler (1889-1945) was an Austrian-born German politicia
assassination in 1945.

Hitler was born in Braunau am Inn, Austria-Hungary, to Alois an
World War I, Hitler dropped out of school and joined the German

In 1918, Hitler became involved with the German Workers' Party
attempted a coup d'état against the Weimar government, known as

After the Nazis came to power in 1933, Hitler became Chancellor
aggressive military expansion. The Nazi regime was responsible

Hitler's leadership style was characterized by his charismatic
```

You can also use python to get started once you have pulled (installed) an llm

Using ollama chat:

```
import ollama from 'ollama'

const response = await ollama.chat({
  model: 'llama2',
  messages: [{ role: 'user', content: 'Why is the sky blue?' }],
})
console.log(response.message.content)
```

For demo, Using a smaller model (1b, 3b parameter model is advised, it will save time)

Assignment (2 hours):

Create a comparative analysis of four language models (RoBERTa, BERT, BART, and GPT) by implementing a text generation benchmark across 7 diverse tasks. Students must programmatically interact with each model, collect their responses, and develop a ranking system based on human evaluation metrics and an automated function.

The final deliverable should include a leaderboard table ranking the models' performance, supported by the metrics as discussed. Use the dataset.txt

## 1. Setup (10 points)
- Successfully load all models
- Prepare environment

## 2. Task Implementation (40 points)
- Implement all 7 tasks
- Process through each model
- Proper error handling

## 3. Metrics Collection (20 points)
- Get the outputs for each model for each task
- Proper documentation of results
- Perform evaluation

## 4. Analysis & Visualization (20 points)
- Create comparisons
- Generate leaderboard
- Analysis of results

## 5. Report & Discussion (10 points)
- Document findings. A short summary of each model, how unique it is in terms of its output.
- Discuss model strengths/weaknesses.
- Suggest improvements.

# How to evaluate?

Student has to evaluate based on the following:

- **relevance**: How well does the response address the prompt?
- **coherence**: Is the text logically structured and well-flowing?
- **creativity**: How creative and innovative is the response?
- **factual_accuracy**: Are any stated facts correct?
- **language_quality**:
    - **grammar**: Grammatical correctness
    - **vocabulary**: Appropriate word choice
    - **style**: Consistent writing style

Programmatic evaluation:

**response_length**: Length of generated text
**vocabulary_diversity**: Unique words / total words
**readability_scores**: *Flesch* reading ease score

---

Note:

The Flesch Reading Ease Score is a readability metric that measures how easy or difficult a text is to read. It was developed by Rudolf Flesch and considers factors like:

- Average sentence length
- Average word length (in syllables)

The formula is:

206.835 - 1.015 × (total words ÷ total sentences) - 84.6 × (total syllables ÷ total words)

The score ranges from 0-100:

- 0-30: Very difficult (College graduate level)
- 30-50: Difficult (College level)
- 50-60: Fairly difficult (10th-12th grade)
- 60-70: Standard (8th-9th grade)
- 70-80: Fairly easy (7th grade)
- 80-90: Easy (6th grade)
- 90-100: Very easy (5th grade)