

# Übungen MMHCI SoSe 2025

# Woche 1: Einstieg

- Installieren Sie Fooocus
  - <https://github.com/lllyasviel/Fooocus>
  - Legen Sie ein Arbeitsverzeichnis focus an
  - In diesem Verzeichnis erzeugen Sie ein virtuelles environment für python3.10 (und >)
    - `python3 -m venv venv`
  - Aktivieren Sie das Environment
    - `source venv/bin/activate`
  - Clonen Sie Fooocus
    - `git clone https://github.com/lllyasviel/Fooocus.git`
    - `cd Fooocus`
  - Starten Sie Fooocus mit
    - `python entry_with_update.py`

# Woche 1: Erzeugen Sie Bilder

- Aufgaben
  - Ein Avatar mit pinkem Haar
  - Ihr Lieblingsessen
  - Einen Lieblingsort
  - Phantasieobjekte Ihrer Wahl
  - Eine Actionfigur (in der Verpackung?)

# Woche 1: Übungen

- Lokale LLMs mit ollama
  - Rufen Sie **ollama** auf und lassen Sie sich die installierten LLMs anzeigen
    - `ollama list`
  - Starten Sie llama3.2
    - `ollama run llama3.2`
- Aufgabe: Finden Sie heraus
  - wer **Alan Turing** war und
  - welchen Bezug er zur KI und
  - weiteren zentralen Themen der Informatik hat

# Woche 2: Übungen

- Interagieren mit LLMs
  - Starten Sie llama3.2 mit ollama
    - `ollama run llama3.2`
- Weitere Modelle
  - Schauen Sie sich in library von Ollama weitere Modelle an
  - Laden Sie einige Modelle herunter und experimentieren Sie mit diesen
  - Laden Sie sich das Vision-Modell von llama3.2-vision herunter
  - Finden Sie Bilder im Netz von Alltagsszenen zu folgenden Szenen
    - Strassenszene in New York
    - Im Schwimmbad
    - Beim Fussball
    - Arbeit und Studium
    - Eine handgeschriebene Einkaufsliste
  - Lassen Sie sich Beschreibungen zu den Bildern generieren
  - Stellen Sie Fragen zu den Bildern

# Woche 2: Übungen

- Sentimentanalyse: Stellen Sie sich vor, dass Sie eine Einschätzung von Studierendenkommentaren zu einer Vorlesung haben möchten. Die Kommentare (Reviews) sollen in 3 Kategorien eingeteilt werden: positiv, neutral, negativ
- Entwickeln Sie einen Prompt, der das LLM dazu bringt, einen übergebenen Review zu analysieren und in eine entsprechende Klasse einzuordnen.
  - Geben Sie im Prompt Beispiele für die jeweiligen Klassen
  - Strukturieren Sie den Prompt (durch Überschriften)
  - Geben Sie dem System Hinweise zur Ordnung der Kategorien
  - Als Ergebnis der Analyse soll lediglich die Kategorie ausgegeben werden
  - Speichern Sie den Prompt in einer Datei „meinePrompts.txt“, der Anfang Ihrer persönlichen Sammlung an guten Prompts

# Woche 2: Übungen

- Programmieren mit LLMs
  - Folgen Sie den Anweisungen für die demo `llm_einbinden` in den Materialien des Kurses
  - Verwenden Sie Ihr eben entwickeltes Prompt und binden Sie dieses in das Programm `llm_sentiment.py` ein
  - Testen Sie das Programm

# Woche 3: chat und TTS

- Basierend auf dem Beispiel **llm\_einbinden.py** erzeugen Sie eine Datei **chat\_ollama.py**, die es erlaubt:
  - Interaktiv mit dem LLM zu reden
- Bauen Sie dazu den Code so um, dass
  - Eine Funktion **ask(question)**: definiert wird, die eine Frage erhält und die Antwort des LLMs zurückgibt
  - Eine Eingabeschleife enthält, die eine Zeile vom Terminal liest und an die Funktion ask(question) übergibt und
    - Anschliessend die Antwort ausgibt
  - Nutzen Sie dazu die Befehle
    - prompt(„Text der vor der Eingabe ausgegeben wird“)
    - input()
    - strip()



# Woche 3: TTS mit say

- Informieren Sie sich über den Befehl **say**, der auf dem Mac zur Verfügung steht
  - Nutzen Sie dazu den Befehl **man**, mit dem Sie das Manual, also die Bedienungsanleitung des Befehls anziehen lassen können
  - Finden Sie heraus, welche deutschen Stimmen es gibt
    - Verwenden Sie dazu den Pipe ( | ) und den Befehl **grep** mit dem Sie die Ausgabe der Hilfe durchsuchen können
  - Binden Sie den say-Befehl in Ihr Python-Programm ein und lassen Sie den Rechner die Antworten des LLM sprechen.