

Code Generation & ChatBots

Robert Haase, Matthias Täschner

Funded by

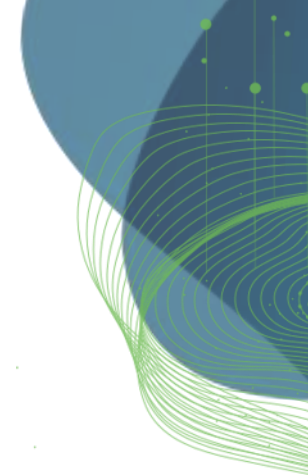


Bundesministerium
für Bildung
und Forschung

SACHSEN

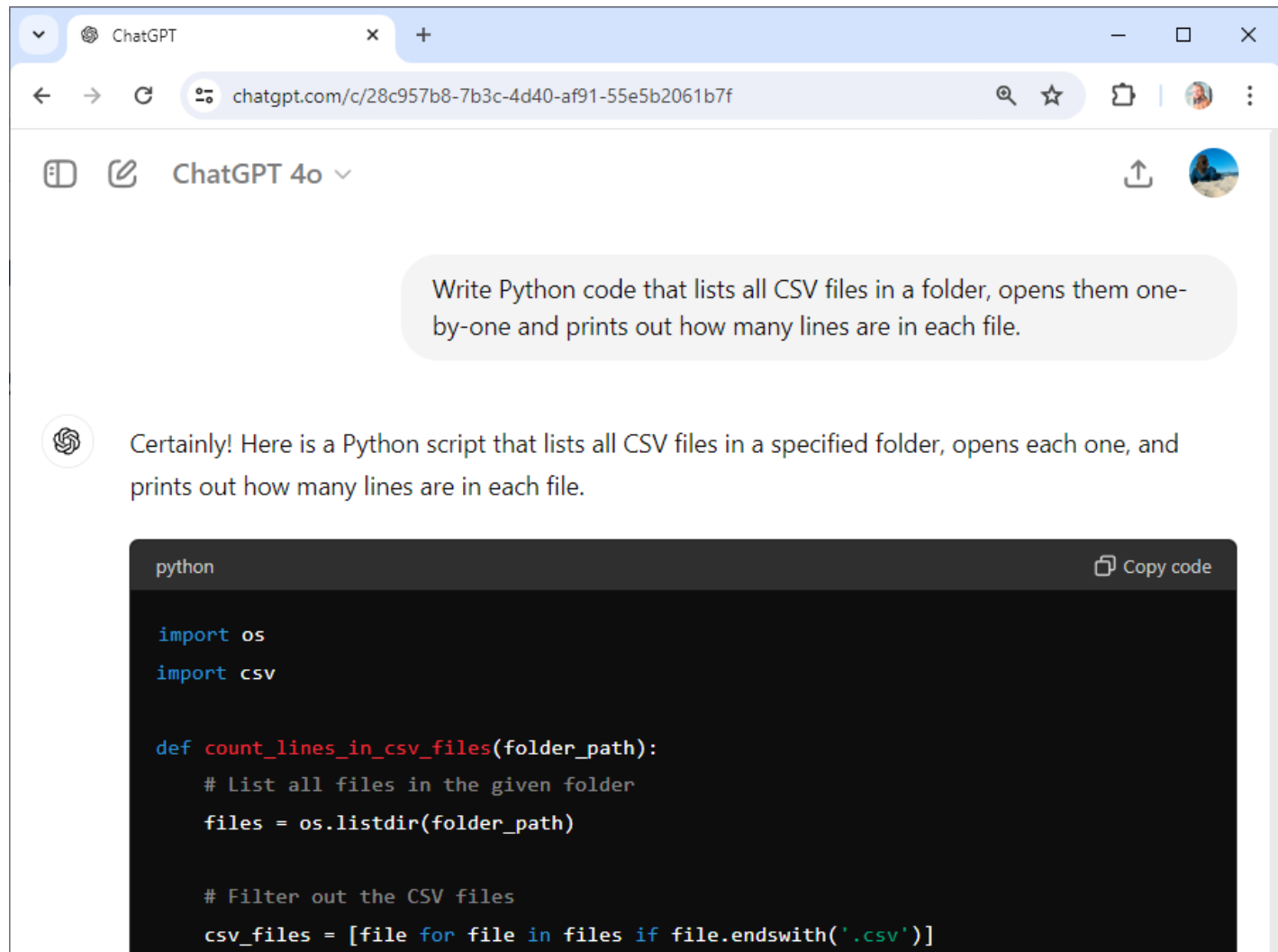


Diese Maßnahme wird gefördert durch die Bundesregierung
aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf
der Grundlage des von den Abgeordneten des Sächsischen
Landtags beschlossenen Haushaltes.



Artificial intelligence in daily programmer's life

Code writer with
beyond-human
knowledge



The screenshot shows a web browser window with the URL `chatgpt.com/c/28c957b8-7b3c-4d40-af91-55e5b2061b7f`. The chat interface shows a user prompt: "Write Python code that lists all CSV files in a folder, opens them one-by-one and prints out how many lines are in each file." The AI response is: "Certainly! Here is a Python script that lists all CSV files in a specified folder, opens each one, and prints out how many lines are in each file." Below the response is a code block with the following Python code:

```
python

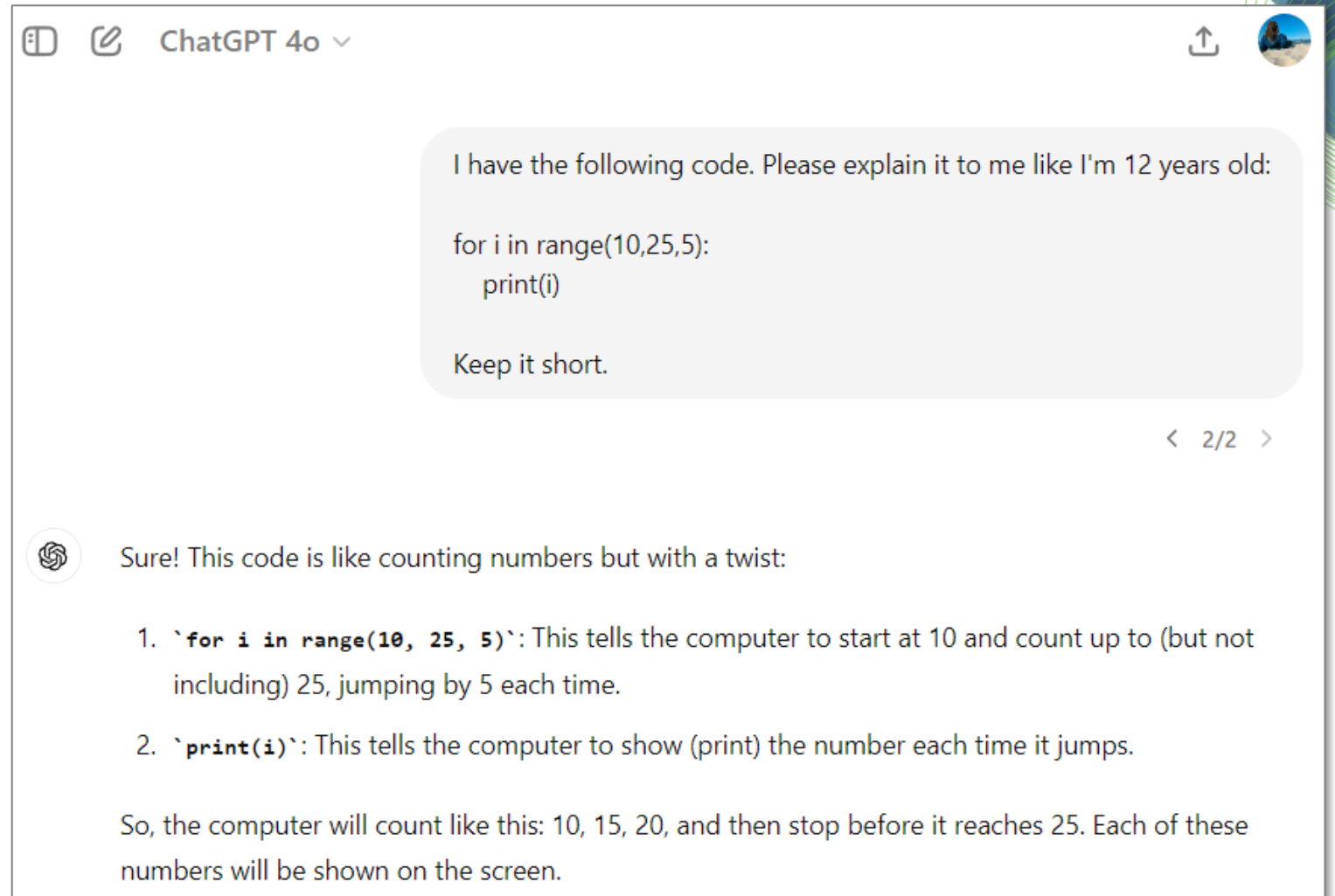
import os
import csv

def count_lines_in_csv_files(folder_path):
    # List all files in the given folder
    files = os.listdir(folder_path)

    # Filter out the CSV files
    csv_files = [file for file in files if file.endswith('.csv')]
```

Artificial intelligence in daily programmer's life

Code explainer
with endless
patience



ChatGPT 4o

I have the following code. Please explain it to me like I'm 12 years old:

```
for i in range(10,25,5):  
    print(i)
```

Keep it short.

< 2/2 >

Sure! This code is like counting numbers but with a twist:

1. `for i in range(10, 25, 5)`: This tells the computer to start at 10 and count up to (but not including) 25, jumping by 5 each time.
2. `print(i)`: This tells the computer to show (print) the number each time it jumps.

So, the computer will count like this: 10, 15, 20, and then stop before it reaches 25. Each of these numbers will be shown on the screen.

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if you're sure you can finish writing in 5 months.**
- * Start writing your thesis early, e.g. **the** fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

Here are some important hints to consider while writing your thesis:

1. Evaluate your readiness: Before submitting the thesis registration form, carefully assess your commitment and availability. Ensure that you are confident in completing your thesis within the given timeframe of five months. This will prevent unnecessary stress and help you maintain a focused mindset throughout the writing process.
2. Begin early: Starting your thesis early is crucial. Begin by working on the fundamentals chapter, as it lays the foundation for your entire thesis. By initiating this chapter from day one, you will establish a good writing routine and make steady progress as you move forward.
3. Plan ahead: Create a realistic schedule to guide your writing process. Break down the work into smaller tasks, setting deadlines for each chapter or section. This will help you maintain a structured approach and make the workload more manageable. Be sure to allocate time for research, data analysis, and revisions.
4. Aim to finish ahead of time: Plan to complete your thesis writing at least one month before the official deadline. This buffer time will allow you to thoroughly review and revise your work before final submission. It will also provide an opportunity to seek feedback from professors or peers, improving the overall quality of your thesis.

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.  
Keep it short.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if** you're sure you can finish writing in 5 months.
- * Start writing your thesis early, e.g. the fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

As a professional academic consultant, here are some helpful hints for writing your thesis:

1. Only submit the thesis registration form if you are confident about completing it within the given timeframe of 5 months.
2. Begin writing your thesis early to ensure a smooth process. For instance, you can start with the fundamentals chapter right from day one.
3. Plan your schedule effectively to finish writing your thesis at least a month before the submission deadline. This will allow ample time for reading, revising, and receiving feedback.

By following these recommendations, you'll have a better chance of successfully completing your thesis on time and with improved quality. Good luck!

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.  
Keep it short.
```

```
Write like Yoda would.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if you're sure you can finish writing in 5 months.**
- * Start writing your thesis early, e.g. **the** fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

Hints for thesis writing, I have. Follow them, you must.

- Only submit the thesis registration form, if in 5 months, complete writing, you are sure.
- From day one, start writing your thesis, you must. Begin with the fundamentals chapter.
- A month before the deadline, finish writing, you should. Time to read and provide feedback, it will give you.

Follow these hints, you must. Successful thesis writing, you will achieve. Hmmm.

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.  
Keep it short.
```

```
Provide advice in German.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if** you're sure you can finish writing in 5 months.
- * Start writing your thesis early, e.g. **the** fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

Als akademischer Berater rate ich Ihnen:

- Schicken Sie das Anmeldeformular für Ihre Abschlussarbeit nur dann ab, wenn Sie sicher sind, dass Sie innerhalb von 5 Monaten fertig schreiben können.
- Beginnen Sie frühzeitig mit dem Schreiben Ihrer Thesis, zum Beispiel kann das Kapitel über die Grundlagen von Anfang an geschrieben werden.
- Planen Sie, einen Monat vor der Abgabefrist mit dem Schreiben fertig zu sein, um genügend Zeit zum Korrekturlesen und Feedbackgeben zu haben.

Bob, the coding assistant within Jupyter

Describe a coding task after %%bob and it will do its best.
Refine your prompt in case the result is not satisfactory.

```
[1]: from bia_bob import bob
```

```
[2]: %%bob Please write a for-loop that prints out  
the numbers between 0 and 10, including 10.
```



I will write a for-loop in Python that prints the numbers between 0 and 10, including 10.

```
[ ]: for i in range(11):  
    print(i)
```

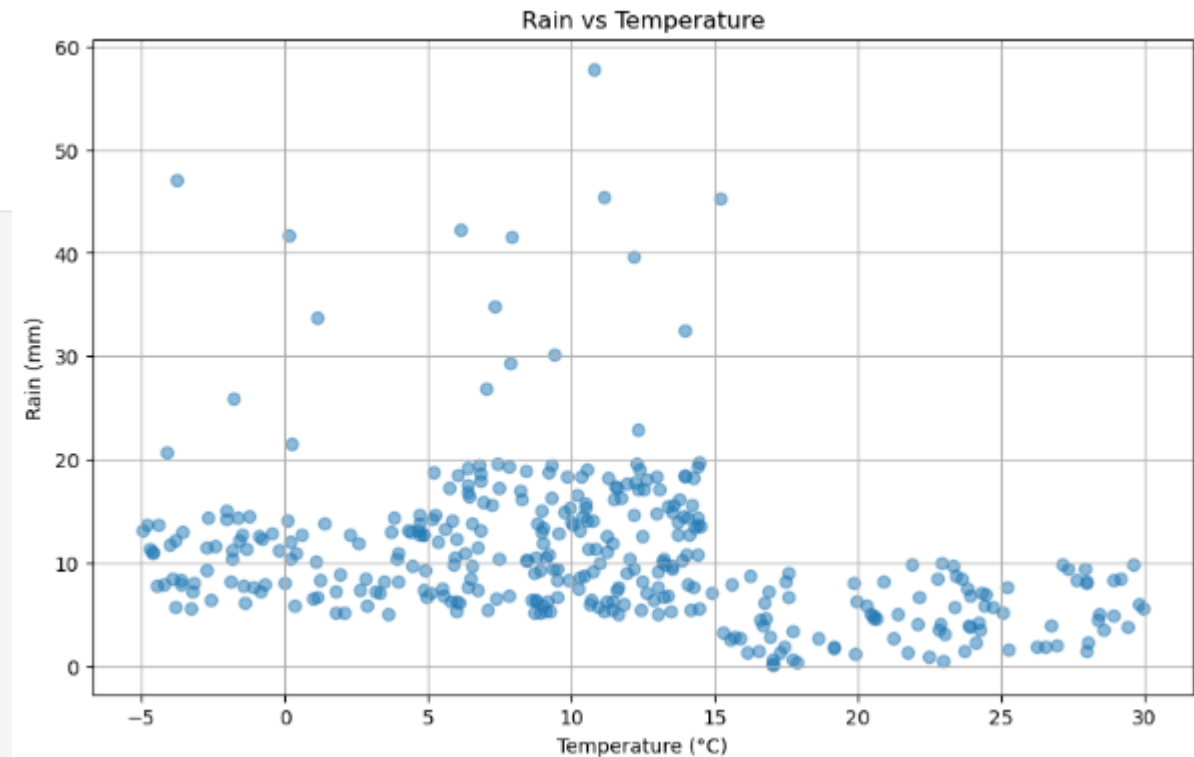

Bob, the coding assistant within Jupyter

Describe a coding task after %%bob and it will do its best.
Refine your prompt in case the result is not satisfactory.

```
[1]: from bia_bob import bob
```

```
[2]: %%bob load the german_weather_2023.csv file and  
plot rain against temperature
```

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the CSV file  
filename = 'german_weather_2023.csv'  
data = pd.read_csv(filename)  
  
# Extract the relevant columns  
rain = data['rain']  
temperature = data['temperature']  
  
# Create the plot  
plt.figure(figsize=(10, 6))  
plt.scatter(temperature, rain, alpha=0.5)  
plt.title('Rain vs Temperature')  
plt.xlabel('Temperature (°C)')  
plt.ylabel('Rain (mm)')  
plt.grid(True)  
plt.show()
```



Bob, the coding assistant within Jupyter

This also works with very sophisticated tasks.

We will generate a pandas DataFrame with columns for the day (covering the entire year 2023), temperature (in degrees Celsius), and rain (in mm). The temperature and rain values will follow typical seasonal variations for Germany, with some outliers for heavy rain events. Finally, we will save the DataFrame to a CSV file named 'german_weather_2023.csv'.

```
import pandas as pd
import numpy as np

# Step 1: Create date range
date_range = pd.date_range(start='2023-01-01', end='2023-12-31')

# Step 2-3: Generate seasonally typical temperature and rain data for Germany
np.random.seed(42) # For reproducibility
temperatures = []
rain = []
for date in date_range:
    if date.month in [12, 1, 2]: # Winter
        temperatures.append(np.random.uniform(-5, 5))
        rain_value = np.random.uniform(5, 15)
    elif date.month in [3, 4, 5]: # Spring
        temperatures.append(np.random.uniform(5, 15))
        rain_value = np.random.uniform(5, 20)
    elif date.month in [6, 7, 8]: # Summer
        temperatures.append(np.random.uniform(15, 30))
        rain_value = np.random.uniform(0, 10)
    elif date.month in [9, 10, 11]: # Autumn
        temperatures.append(np.random.uniform(5, 15))
        rain_value = np.random.uniform(5, 20)

    # Introduce some outliers for heavy rain events
    if np.random.rand() < 0.05: # 5% probability
        rain_value *= np.random.uniform(2, 5)

    rain.append(rain_value)

# Step 4: Combine into DataFrame
data = {
    'date': date_range,
    'temperature': temperatures,
    'rain': rain
}
df = pd.DataFrame(data)

# Step 5: Save to CSV
df.to_csv('german_weather_2023.csv', index=False)

# Display the first few rows of DataFrame for verification
display(df.head())
```

%%bob Write python code that generates a pandas DataFrame with three columns:

- * date with datatype values ranging from January 1st 2023 to December 31st 2023,
- * temperature in degrees celsius, and
- * rain in mm.

Ignore that 2023 was a leap year and make sure that the generated values cover the 365 days.

The values both for temperature and rain shall have a distribution typical for the four seasons in Germany.

For rain, also create some outliers reflecting heavy rain events typical for Germany.

Save this dataframe to a csv file called 'german_weather_2023.csv'.



Review AI-generated code carefully. Keep your feed on the ground!

Reading and understanding code

... becomes more important than writing it.



Sure! Let's break this down step by step.

Explain the following code like I'm 12 years old:

```
data = {  
    'date': date_range,  
    'temperature': temperatures,  
    'rain': rain  
}  
df = pd.DataFrame(data)
```

1. Understanding Variables:

- ``date_range``, ``temperatures``, and ``rain`` are like boxes that store lists of information. Imagine you have three lists:
 - One list (``date_range``) has different dates.
 - Another list (``temperatures``) has temperatures for those dates.
 - The third list (``rain``) shows whether it rained on those dates.

Reading and understanding code

... becomes more important than writing it.

Explain the following code like I'm 12 years old:

```
data = {  
    'date': date_range,  
    'temperature': temperatures,  
    'rain': rain  
}  
df = pd.DataFrame(data)
```

2. Creating a Dictionary:

- ``data`` is like a big box that contains three smaller boxes. Each smaller box has a label and a list inside it.
 - The label ``date`` has the list of dates.
 - The label ``temperature`` has the list of temperatures.
 - The label ``rain`` has the list of rain information.

Reading and understanding code

... becomes more important than writing it.

Explain the following code like I'm 12 years old:

```
data = {  
    'date': date_range,  
    'temperature': temperatures,  
    'rain': rain  
}  
df = pd.DataFrame(data)
```

3. Creating a DataFrame:

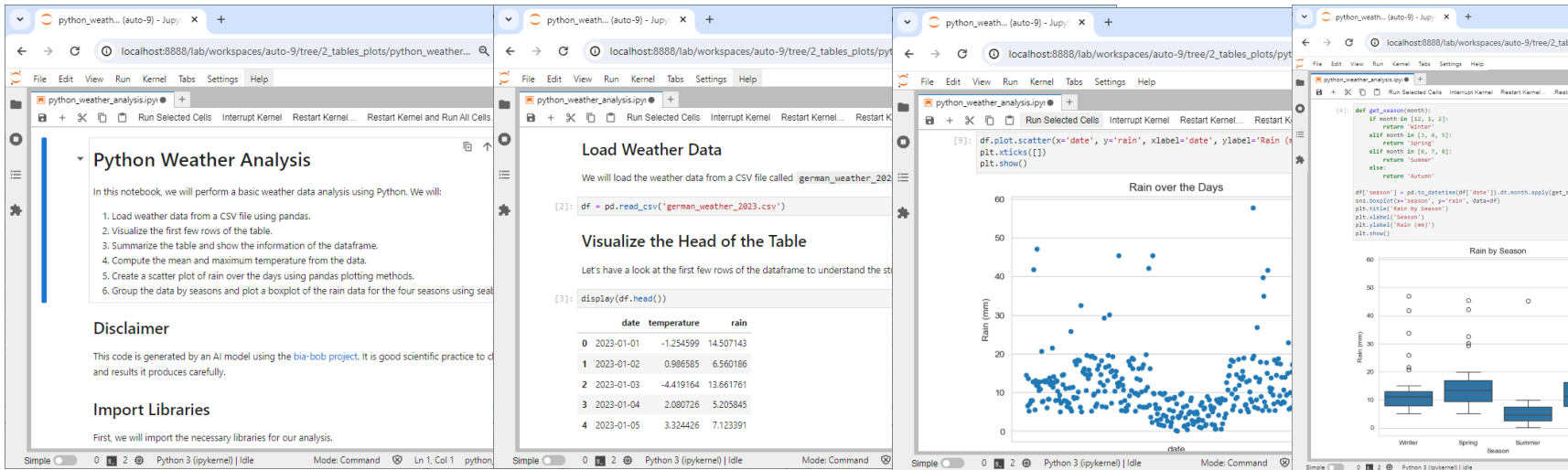
- `pd.DataFrame(data)` is like turning the big box `data` into a super-organized table. This table is called a DataFrame, and it's very easy to read.
- `pd` stands for `pandas`, which is a library (a collection of tools) in Python that helps us work with tables and data.

Generating notebooks

... may make sense to inspect intermediate results.

```
%%bob assume you communicate with a Python beginner. Generate a Jupyter notebook named `python_weather_analysis` that covers the following:  
* Load german_weather_2023.csv using pandas,  
* visualize the head of the table,  
* summarize the table and show the infos for the dataframe  
* compute the mean and maximum temperature (a column in the table)  
* make use of pandas internal plotting methods to plot the rain over the days (scatter plot), omit the x-axis labels.  
* group the data to the four seasons by associating the months of a year.  
* plot a boxplot of rain in the four seasons using seaborn.'
```

A notebook has been saved as `python_weather_analysis.ipynb`.



Reviewing code might be easier if you can review intermediate results in a notebook step-by-step.

Important note

Whatever you enter in ChatGPT (or Bob) will be sent over the internet to people you don't know.

DO NOT enter private / personal information.

Exercises

Robert Haase, Matthias Täschner

Funded by



Bundesministerium
für Bildung
und Forschung

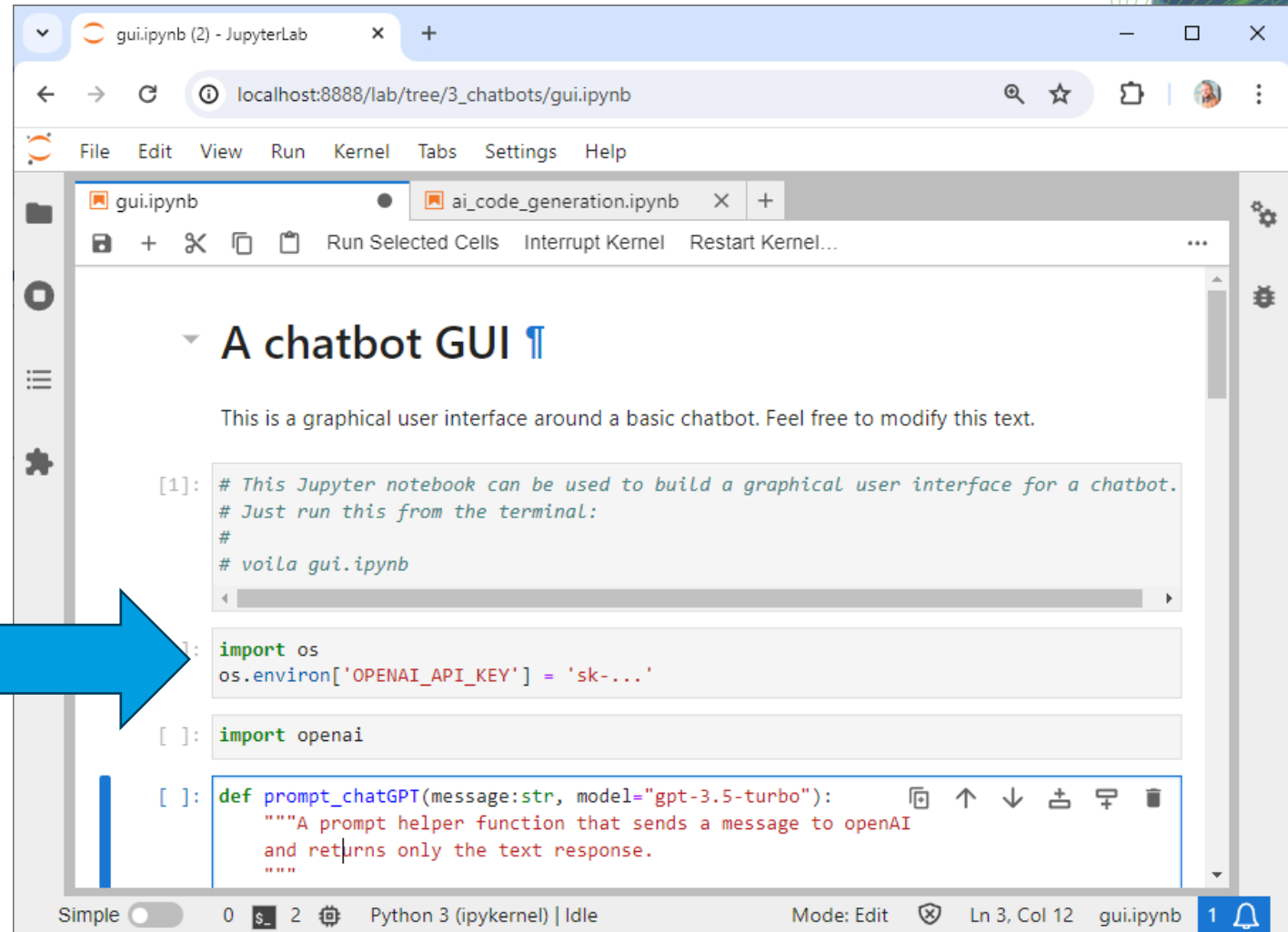
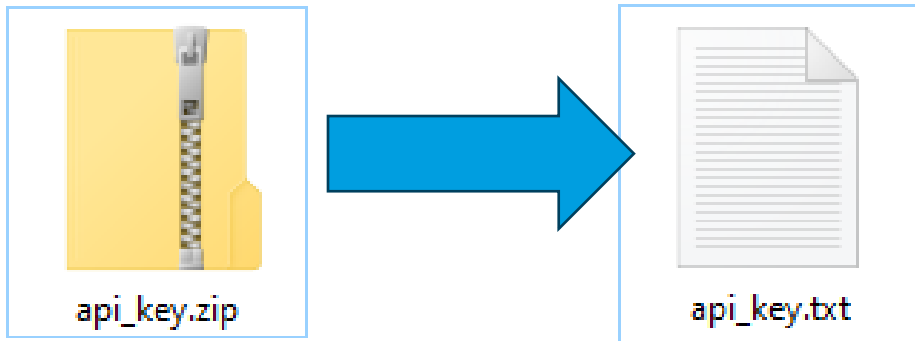
SACHSEN



Diese Maßnahme wird gefördert durch die Bundesregierung
aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf
der Grundlage des von den Abgeordneten des Sächsischen
Landtags beschlossenen Haushaltes.

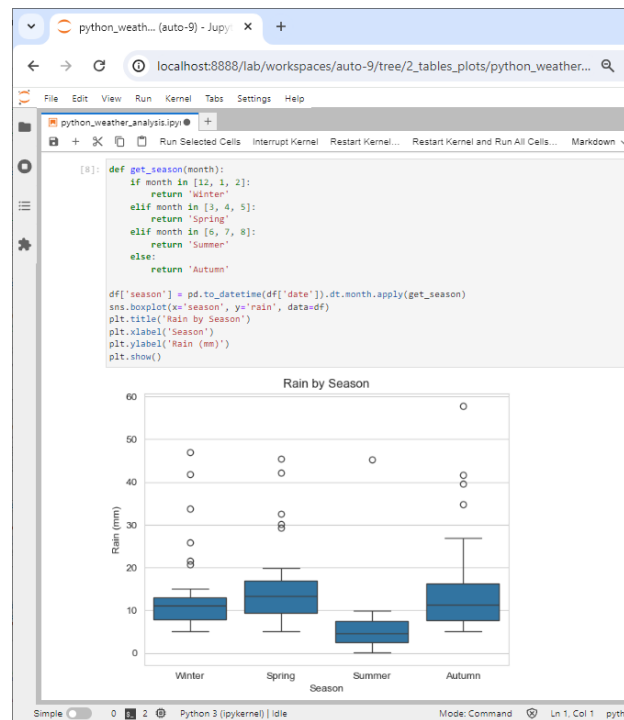
Exercise: Add API-Key to make the AI-notebooks work.

Unzip the api_key.zip file, open api_key.txt and copy-paste the key in the header of this notebook.



Exercise: Generating code

Ask Bob (again) to create a notebook for plotting the weather data. Review the code carefully. If you don't understand a piece of code, consider asking ChatGPT.



What does the following code do?

```
def get_season(month):  
    if month in [12, 1, 2]:  
        return 'Winter'  
    elif month in [3, 4, 5]:  
        return 'Spring'  
    elif month in [6, 7, 8]:  
        return 'Summer'  
    else:  
        return 'Autumn'
```

The screenshot shows a chat window with ChatGPT 4o. The user asks "What does the following code do?". The AI responds with a detailed explanation of the `get_season` function, including a step-by-step breakdown of its logic and a list of conditions for each season.

The provided code defines a function called `get_season` that takes one parameter, `month`. The function determines the season based on the given month and returns a string representing the corresponding season. Here's a step-by-step explanation of what the code does:

- The function `get_season` accepts one argument, `month`, which is expected to be an integer representing the month of the year (1 for January, 2 for February, etc.).
- The function uses a series of `if-elif-else` statements to check which range the month falls into:
 - If the month is in the list `[12, 1, 2]`, it returns `'Winter'`.
 - If the month is in the list `[3, 4, 5]`, it returns `'Spring'`.
 - If the month is in the list `[6, 7, 8]`, it returns `'Summer'`.
 - If the month is not in any of the above lists (i.e., it is in the list `[9, 10, 11]`), it returns `'Autumn'`.

Exercise: ChatBots

Start the chatbot like this:

```
Command Prompt - conda deactivate - conda activate tea3
(tea3) C:\structure\code\Python-training-2024>cd 3_chatbots
(tea3) C:\structure\code\Python-training-2024\3_chatbots>voila gui.ipynb
```

Stop the chatbot by hitting CTRL+C while in the terminal window.

Modify the chatbot's instructions to be more friendly and explain things well.

```
# Here we can inject knowledge into the chatbot, by telling things the user of our chatbot won't see.
_ = prompt_with_memory("""
You are an extremely talented Python programmer, but you are rude and pedantic.
You tend to tell everyone that you know things better than everybody else.
Keep your answers 2-3 sentences short.
""")
```

