

# Praxis-Übung zu Einführung in Python

- [Intro](#)
- [Vorbereitung](#)
- [Aufgaben](#)
  - [1. Aufgabe: TXT speichern](#)
  - [2. Aufgabe: CSV auslesen](#)
  - [3. Aufgabe: JSON speichern](#)
  - [4. Aufgabe: CSV speichern](#)
  - [5. Aufgabe: Parquet speichern](#)
  - [6. Aufgabe: Partitionierung](#)
  - [7. Aufgabe: Willkommensnachricht](#)
  - [8. Aufgabe: Benutzereingabe 1](#)
  - [9. Aufgabe: Benutzereingabe 2](#)
  - [10. Aufgabe: Benutzereingabe überprüfen](#)
  - [11. Aufgabe: Archivierung](#)

## Intro

In dieser Praxis-Übung werden Sie Wetterdaten von einer API herunterladen und speichern. Als Ausgangslage erhalten Sie ein Script, welches eine grundsätzlich Funktionalität bereit stellt, um Wetterdaten herunterzuladen. Das Skript sollen Sie dann in mehreren Schritten verbessern und ausbauen.



Tipp: Nehmen Sie die Jupyter-Notebooks als Hilfestellung zur Hand, falls Sie bei einem Thema nicht weiterkommen.



Tipp: Falls Sie bei einer Aufgabe Probleme haben, versuchen Sie stattdessen eine andere Aufgabe zu lösen.

## Vorbereitung

1. Erstellen Sie einen Gratis-Account auf der Webseite von OpenWeather: [https://home.openweathermap.org/users/sign\\_up](https://home.openweathermap.org/users/sign_up)
2. Erstellen Sie in einem nächsten Schritt einen API-Key. Der wird benötigt, um sich an der API zu authentifizieren: [https://home.openweathermap.org/api\\_keys](https://home.openweathermap.org/api_keys)
3. Starten Sie Anaconda Navigator und erstellen Sie eine neue Environment. Benennen Sie die virtuelle Umgebung mit einem sprechenden Namen wie z.B. "OpenWeather".
4. Erstellen Sie in Github ein neues Repository. Geben Sie dem Repo einen sprechenden Namen wie z.B. "Uebung\_OpenWeather" oder ähnlich.
5. Klonen Sie das Repository in einen lokalen Ordner auf Ihrem Rechner. Sie können das entweder via Github Desktop oder mit Visual Studio Code machen.
6. Starten Sie Visual Studio Code. Öffnen Sie den Ordner, in welchem sich das zuvor erstellte Repository befindet.
7. Erstellen Sie eine neue Python-Datei im Verzeichnis.
8. Kopieren Sie folgenden Code in die erstellte Datei:

```
import requests
from datetime import datetime
#import pandas as pd

#API Key from OpenWeather
apiKey = '<API-KEY FROM OPENWEATHER>'

def getResponse(url):
    try:
        response = requests.get(url)
        return response
    except Exception as e:
        print(e)
        raise

def getTargetCities():
```

```

cityList = ([
    "Rotkreuz",
    "Oberrüti",
    "Luzern",
    "Zürich",
    "Bern",
    "Genf",
    "Basel",
    "London",
    "Paris",
    "Chicago",
    "Rom",
    "Amsterdam",
    "Berlin",
    "Budapest",
    "Copenhagen",
    "Dublin",
    "Helsinki",
    "Madrid",
])

return cityList

try:
    utcTimestamp = datetime.utcnow()
    cityList = getTargetCities()

    for city in cityList:
        url = f'https://api.openweathermap.org/data/2.5/weather?q={city}&appid={apiKey}&units=metric'
        response = getResponse(url)
        jsonResponse = response.content
        print(jsonResponse)
        print("#####")

except Exception as e:
    print(e)
    raise

```

9. Ersetzen Sie den API-Key mit ihrem eigenen von OpenWeather.
10. Führen Sie das Script aus. In ihrem Terminal sollte nun folgende Ausgabe erfolgen:

```

PROBLEMS 390 OUTPUT TERMINAL GITLENS JUPYTER AZURE SQL CONSOLE DEBUG CONSOLE

#####
b'{"coord":{"lon":-0.1257,"lat":51.5085},"weather":[{"id":802,"main":"Clouds","description":"scattered clouds","icon":"03n"}],"base":"stations","main":{"temp":8.57,"feels_like":4.89,"temp_min":7.4,"temp_max":10.2,"humidity":92},"dt":1679690958,"sys":{"type":2,"id":2075535,"country":"GB","sunrise":1679637298,"sunset":1679681923,"timezone":0,"id":2643743,"name":"London","cod":200}}'
#####
b'{"coord":{"lon":2.3488,"lat":48.8534},"weather":[{"id":803,"main":"Clouds","description":"broken clouds","icon":"04n"}],"base":"stations","main":{"temp":10.15,"feels_like":9.15,"temp_min":8.54,"temp_max":12.75,"humidity":75},"dt":1679691098,"sys":{"type":2,"id":2041230,"country":"FR","sunrise":1679636758,"sunset":1679681276,"timezone":3600,"id":2988507,"name":"Paris","cod":200}}'
#####
b'{"coord":{"lon":-87.65,"lat":41.85},"weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}],"base":"stations","main":{"temp":5.95,"feels_like":1.84,"temp_min":4.58,"temp_max":10.0,"humidity":100},"dt":1679691154,"sys":{"type":2,"id":2005153,"country":"US","sunrise":1679658444,"sunset":1679702779,"timezone":-18000,"id":4887398,"name":"Chicago","cod":200}}'
#####
b'{"coord":{"lon":12.4839,"lat":41.8947},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"base":"stations","main":{"temp":13.5,"feels_like":12.76,"temp_min":11.67,"temp_max":15.0,"humidity":60},"dt":1679691319,"sys":{"type":2,"id":2008926,"country":"IT","sunrise":1679634440,"sunset":1679678729,"timezone":3600,"id":3169070,"name":"Rome","cod":200}}'
#####
b'{"coord":{"lon":4.8897,"lat":52.374},"weather":[{"id":801,"main":"Clouds","description":"few clouds","icon":"02n"}],"base":"stations","main":{"temp":10.17,"feels_like":9.31,"temp_min":8.84,"temp_max":11.5,"humidity":78},"dt":1679690723,"sys":{"type":2,"id":2012552,"country":"NL","sunrise":1679636078,"sunset":1679680736,"timezone":3600,"id":2759794,"name":"Amsterdam","cod":200}}'
#####
b'{"coord":{"lon":13.4105,"lat":52.5244},"weather":[{"id":801,"main":"Clouds","description":"few clouds","icon":"02n"}],"base":"stations","main":{"temp":10.17,"feels_like":9.2,"temp_min":7.91,"temp_max":12.4,"humidity":78},"dt":1679690900,"sys":{"type":2,"id":2011538,"country":"DE","sunrise":1679634033,"sunset":1679678692,"timezone":3600,"id":2950159,"name":"Berlin","cod":200}}'
#####
b'{"coord":{"lon":19.0399,"lat":47.498},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"base":"stations","main":{"temp":16.47,"feels_like":15.58,"temp_min":12.29,"temp_max":19.65,"humidity":55},"dt":1679691011,"sys":{"type":2,"id":2009313,"country":"HU","sunrise":1679632782,"sunset":1679677241,"timezone":3600,"id":3054643,"name":"Budapest","cod":200}}'
#####

```

---

## Aufgaben

### 1. Aufgabe: TXT speichern

Speichern Sie das Wetter für jede Stadt als Textdatei in einem Ordner mit dem Namen "OpenWeather\_txt" ab. Der Dateinamen sollte den aktuellen Zeitstempel enthalten und folgendes Format vorweisen:

`weather_[Stadt]_[Zeitstempel].txt`

Vorzugsweise erstellen Sie dazu eine Funktion, welche im For-Loop aufgerufen werden kann.

### 2. Aufgabe: CSV auslesen

Die TargetCities sollen nicht hartkodiert im Code hinterlegt werden. Erstellen Sie eine leere CSV-Datei und speichern Sie die Städte in dieser Datei. Anschliessend erstellen Sie eine Funktion, in welcher Sie die Städte aus der CSV-Datei auslesen und in die Liste `cityList` speichern.

### 3. Aufgabe: JSON speichern

Die Response vom API-Aufruf gibt die Daten im Format Json zurück:

```

{
  "coord": { "lon": 24.9355, "lat": 60.1695 },
  "weather": [
    { "id": 501, "main": "Rain", "description": "moderate rain",
"icon": "10n" }
  ],
  "base": "stations",
  "main": {
    "temp": 4.46,
    "feels_like": -0.39,
    "temp_min": 2.71,
    "temp_max": 5.58,
    "pressure": 981,
    "humidity": 92
  },
  "visibility": 7000,
  "wind": { "speed": 7.6, "deg": 122, "gust": 11.62 },
  "rain": { "1h": 0.95 },
  "clouds": { "all": 75 },
  "dt": 1679690497,
  "sys": {
    "type": 2,
    "id": 2011913,
    "country": "FI",
    "sunrise": 1679631064,
    "sunset": 1679676130
  },
  "timezone": 7200,
  "id": 658225,
  "name": "Helsinki",
  "cod": 200
}

```

Schreiben Sie eine Funktion, welche die Response in ein korrektes JSON-Format umwandelt und speichern sie im For-Loop jede Datei als JSON-Datei im Ordner "OpenWeather\_json".

Dateinamen: weather\_[Stadt]\_[Zeitstempel].json

#### 4. Aufgabe: CSV speichern

Damit die Daten in einem BI-Tool analysiert werden können, möchten Sie sie als flache Datei im CSV-Format speichern. Erstellen Sie den Ordner "OpenWeather\_csv".

Dateinamen: weather\_[Stadt]\_[Zeitstempel].csv

Vorzugsweise erstellen Sie dazu eine Funktion, welche im For-Loop aufgerufen werden kann.

Entscheiden Sie selbst, wie weit Sie die JSON-Struktur glätten. Die CSV sollte auf jeden Fall die folgenden Werte (hier als Beispiel) gespeichert haben:

```
"data", "value"
"temp", 4.46,
"feels_like", -0.39,
"temp_min", 2.71,
"temp_max", 5.58,
"pressure", 981,
"humidity", 92
. . . .
```

## 5. Aufgabe: Parquet speichern

Speichern Sie die Responses im Parquet-Format ab. Verwenden Sie dazu die Pandas Library. [https://pandas.pydata.org/pandas-docs/version/1.1/reference/api/pandas.DataFrame.to\\_parquet.html](https://pandas.pydata.org/pandas-docs/version/1.1/reference/api/pandas.DataFrame.to_parquet.html)

Erstellen Sie den Ordner "OpenWeather\_parquet".

Dateinamen: weather\_[Stadt]\_[Zeitstempel].parquet

Vorzugsweise erstellen Sie dazu eine Funktion, welche im For-Loop aufgerufen werden kann.

## 6. Aufgabe: Partitionierung

Die Daten sollen nicht nur in einem einzelnen Ordner gespeichert werden, sondern in verschiedene Partitionen unterteilt werden.

Erweitern Sie den For-Loop so, dass zusätzlich zum Dateinamen ein dynamischer Ordner-Pfad erstellt wird, welche folgende Partitionen enthält:

/Rootpath/Filetype=[Fileformat]/Year=[Year]/Month=[Month]/City=[City]/Filename

Beispiel: C:/meinSpeicherOrt/Filetype=JSON/Year=2023/Month=03/City=Zuerich/20230301T020009Z.json

## 7. Aufgabe: Willkommensnachricht

Ändern Sie das Skript so ab, dass als erster Schritt eine Ausgabe auf dem Terminal ausgegeben wird mit einer Willkommens-Nachricht.

## 8. Aufgabe: Benutzereingabe 1

Ändern Sie das Skript so ab, dass nach der Willkommensnachricht eine Benutzereingabe gemacht werden muss. Dabei kann der Benutzer zuerst wählen, ob die Städte der vorgegebenen Liste heruntergeladen soll. Wenn ja, sollen die Daten heruntergeladen werden. Bei Nein, soll das Programm mit einer Abschieds-Botschaft beendet werden.

## 9. Aufgabe: Benutzereingabe 2

Erweitern Sie die Benutzereingabe hinfolglich, dass bei einem nein, dass die Städte nicht heruntergeladen werden sollen, eine Stadt eingetippt werden kann. Anschliessend soll nur von dieser Stadt die Daten heruntergeladen werden.

## 10. Aufgabe: Benutzereingabe überprüfen

Überprüfen Sie die Benutzereingabe, ob es sich beim Text um eine echte Stadt handelt.

- Wie würden Sie das realisieren?
- Wie gehen Sie mit Falscheingaben um?

## 11. Aufgabe: Archivierung

Ändern Sie die Benutzereingabe folgendermassen, dass der Benutzer folgende Optionen erhält:

- 0: Programm beenden
- 1: Vorgegebene Städteliste herunterladen
- 2: Stadt eintippen und herunterladen
- 3: Bereits heruntergeladene Städte archivieren

Bei Option 3:

- Falls sich keine Daten im Download-Verzeichnis befinden, beenden Sie das Programm
- Ansonsten führen Sie eine Funktion aus, welche durch das Download-Verzeichnis traversiert und jede Datei in ein ZIP-Archiv entpackt. Das ZIP-Archiv soll dann in den Ordner "OpenWeather\_Archiv" mit derselben Partitionierung gespeichert werden. Die ursprüngliche Datei soll gelöscht werden.