* **Github**
* **Repo**  
  <https://github.com/arakonon/Wetterstation>
* **Python-Grundlagen & OOP**
* Python-Kurs.eu: „Objektorientierte Programmierung in Python“  
  <https://www.python-kurs.eu/python_OOP.php>  
  → Basis für Klassenaufbau und -struktur in allen .py-Dateien
* **Auswertung.py**
* Wie man CSV-Dateidaten in Pandas visualisiert und in Diagramme oder Plots umwandelt  
  <https://de.ittrip.xyz/python/pandas-csv-data-visualization>
* Pandas User Guide – Visualization  
  <https://pandas.pydata.org/docs/user_guide/visualization.html>  
  → Einlesen von CSV, Zeitreihen-Index und Subplots mit Matplotlib
* **UvApi.py (Klasse UvApiClient)**
* OpenUV API Documentation  
  <https://www.openuv.io/documentation>
* Python's Requests Library (Guide)  
  https://realpython.com/python-requests/  
  → URL-Aufbau und Header („x-access-token“) für UV-Abfrage
* **Mqtt.py (Klassen EspAußen & MQTTPublisher)**
* tutorials-raspberrypi.de: „Datenaustausch Raspberry Pi – MQTT Broker & Client“  
  <https://tutorials-raspberrypi.de/datenaustausch-raspberry-pi-mqtt-broker-client/>
* MQTT in Python with Paho Client: Beginner's Guide 2025  
  https://www.emqx.com/en/blog/how-to-use-mqtt-in-python
* Random Nerd Tutorials: „ESP32 MQTT Publish/Subscribe (Arduino IDE)“  
  <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>  
  → Aufbau des Paho-Clients, on\_connect/on\_message, publish/subscribe
* **lcd.py (Klassen LcdControl & lcdCheck)**
* RPLCD – Python-Bibliothek für Character-LCDs  
  <https://rplcd.readthedocs.io/en/stable/>
* joy-it SBC-LCD16×2 Anleitung  
  <https://joy-it.net/files/files/Produkte/SBC-LCD16x2/SBC-LCD16x2_Anleitung_2024-03-13.pdf>  
  → Initialisierung, Custom‐Chars, Button-Entprellung im Thread
* **Datenbank.py (Klasse Datenbank)**
* Python-Kurs.eu: CSV-Dateien lesen & schreiben (Kapitel über csv-Modul)  
  <https://www.python-kurs.eu/pandas_data_files.php>
* csv — CSV File Reading and Writing  
  https://docs.python.org/3/library/csv.html  
  → Puffern, Mittelwertbildung, Header-Handling, Pathlib
* **Buzzer.py / Ampel.py**
* joy-it COM-KY012APB Anleitung  
  <https://joy-it.net/files/files/Produkte/COM-KY012APB/COM-KY012APB_Anleitung_2024-05-06.pdf>
* gpiozero Documentation  
  <https://gpiozero.readthedocs.io/en/stable/>  
  → PWM-Buzzersteuerung mit gpiozero
* **BME680.py (Klasse BME680)**
* BoschSensortec: BSEC Arduino-Library (GitHub)  
  <https://github.com/BoschSensortec/BSEC-Arduino-library>
* pi3g: bme68x-python-library und Dokumentation (GitHub)  
  <https://github.com/pi3g/bme68x-python-library/blob/main/(Documentation.md)>  
  → BSEC-Initialisierung, Sampling, State-Save/Load
* BME680 Datasheet  
  https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme680-ds001.pdf
* Interfacing BME680  
  https://www.circuitschools.com/interfacing-bme680-with-arduino-also-measure-indoor-air-quality-index/
* **DHT\_Unified\_Sensor.ino (ESP32-Außenstation)**
* joy-it SEN-DHT22 Anleitung  
  [https://joy-it.net/files/files/Produkte/SEN-DHT22/SEN-DHT22\_Anleitung\_2024-04-11.pdf](https://joy-it.net/files/files/Produkte/SEN-DHT22/SEN-DHT22_Anleitung_2024-04-11.pdf" \t "_new)
* joy-it SEN-UV01  
  <https://joy-it.net/de/products/SEN-UV01>
* A Practical Guide To ESP32 Deep Sleep Modes  
  https://www.programmingelectronics.com/esp32-deep-sleep-mode/
* Random Nerd Tutorials: ESP32 Deep Sleep & Wake-Up Sources  
  <https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/>
* Random Nerd Tutorials: ESP32 MQTT Publish/Subscribe  
  <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>
* WiFi Web Client  
  <https://docs.arduino.cc/retired/library-examples/wifi-library/WiFiWebClient/>
* ESP32 ADC – Read Analog Values with Arduino IDE  
  <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>
* **Bugfixing, Problemlösung, Denkanstöße + Burn in Code**
* ChatGPT  
  <https://chatgpt.com>
* Burn in Code Chat: siehe beigelegte Datei (WetterstationProjekt/Burn\_in\_Code + Chat/ChatGPT\_BurnIn\_Chat.txt)