

1 | with und open()

- with schließt Datei zuverlässig.
- encoding='utf-8' explizit setzen.

```
with open("notiz.txt", "r", encoding="utf-8") as f:
    text = f.read()
```

2 | Dateimodi

Modus	Bedeutung
r	lesen
w	neu schreiben/überschreiben
a	anhängen
x	nur neu erstellen
b	Binärmodus ergänzen

3 | Text schreiben

- w überschreibt vorhandene Datei.
- Zeilenende bewusst ergänzen.

```
with open("log.txt", "w", encoding="utf-8") as f:
    f.write("Start\n")
    f.write("Messung OK\n")
```

4 | Zeilen lesen

- Direktes Iterieren ist speichersparend.
- strip entfernt das Zeilenende und Rand-Leerraum.

```
with open("werte.txt", encoding="utf-8") as f:
    for zeile in f:
        wert = zeile.strip()
        print(wert)
```

5 | pathlib

- Objektorientierte Pfade, plattformunabhängig.
- exists/is_file prüfen Ziel.

```
from pathlib import Path
pfad = Path("daten") / "werte.txt"
if pfad.exists():
    print(pfad.read_text(encoding="utf-8"))
```

6 | Pfade relativ/absolut

- Relativpfad bezieht sich auf Arbeitsverzeichnis.
- __file__ kann Skriptposition liefern.
- Keine Windows-Backslashes unbedacht schreiben.
- Pfade mit pathlib zusammensetzen.
- Zielordner vor Schreiben prüfen.

7 | Ordner anlegen

```
from pathlib import Path
ordner = Path("ausgabe")
ordner.mkdir(parents=True, exist_ok=True)
datei = ordner / "bericht.txt"
```

8 | Datei sicher prüfen

- Fehlende Datei gezielt behandeln.
- Fehler nicht pauschal verschlucken.

```
try:
    text = pfad.read_text(encoding="utf-8")
except FileNotFoundError:
    print(f"Fehlt: {pfad}")
```

9 | Liste von Dateien

```
from pathlib import Path
for datei in Path("daten").glob("*.csv"):
    print(datei.name)
```

10 | Binärdateien

- Bilder/PDFs im Modus rb/wb.
- Nicht als Text dekodieren.

```
with open("bild.png", "rb") as f:
    daten = f.read()
print(len(daten))
```

11 | Typische Fehler

Problem	Prüfen
Datei fehlt	Arbeitsordner/Pfad
Umlaute kaputt	UTF-8
Inhalt weg	w statt a
PermissionError	Rechte/Ziel
Backslash-Fehler	pathlib/raw string

12 | Mini-Übung: Messprotokoll

Aufgabe

Schreibe drei Messwerte zeilenweise in eine Datei.

Musterlösung

```
werte = [21.5, 22.0, 21.8]
with open("messung.txt", "w", encoding="utf-8") as f:
    for wert in werte:
        f.write(f"{wert}\n")
```