

## 1 | Modul importieren

- Modul stellt Funktionen und Konstanten bereit.
- Namen über modul.name verwenden.

```
import math
print(math.pi)
print(math.sqrt(81))
```

## 2 | Gezielt importieren

- Nur benötigte Namen importieren.
- Wildcard-Import vermeiden.

```
from pathlib import Path
from math import pi, sqrt

pfad = Path("daten.txt")
```

## 3 | Alias

- Kurzer, etablierter Alias kann Lesbarkeit verbessern.
- Keine überraschenden Umbenennungen.

```
import statistics as stats
mittel = stats.mean([2, 4, 6])
```

## 4 | Eigenes Modul

- Funktionen in separate .py-Datei auslagern.
- Dateiname ohne Leerzeichen/Bindestrich.

```
# berechnung.py
def flaeche(a, b):
    return a * b

# main.py
import berechnung
```

## 5 | \_\_name\_\_

```
def main():
    print("Programmstart")

if __name__ == "__main__":
    main()
```

## 6 | Standardbibliothek

- math: Mathematik.
- pathlib: Dateipfade.
- datetime: Datum/Zeit.
- csv/json: strukturierte Daten.
- statistics: einfache Statistik.
- Vor Installation zuerst Standardbibliothek prüfen.

## 7 | Paket installieren

```
python -m pip install paketname
python -m pip show paketname
python -m pip uninstall paketname
```

## 8 | Virtuelle Umgebung

- Trennt Abhängigkeiten pro Projekt.
- Im Projektordner erzeugen, nicht ins Git aufnehmen.

```
python -m venv .venv
# Windows:
.venv\Scripts\activate
# Linux/macOS:
source .venv/bin/activate
```

## 9 | Interpreter prüfen

- IDE muss Interpreter der venv nutzen.
- which/where zeigt aktives Python.

```
python --version
python -m pip --version
# Windows: where python
# Linux/macOS: which python
```

## 10 | requirements.txt

```
python -m pip freeze > requirements.txt
python -m pip install -r requirements.txt
```

## 11 | Projektstruktur

```
mein_projekt/
- .venv/
- src/
  - main.py
  - berechnung.py
- tests/
- requirements.txt
```

## 12 | Mini-Übung: Modul

### Aufgabe

Lege ein Modul umrechnung.py mit mm\_in\_m() an und importiere es.

### Musterlösung

```
# umrechnung.py
def mm_in_m(mm):
    return mm / 1000

# main.py
from umrechnung import mm_in_m
print(mm_in_m(500))
```