

Programmieren – 7. Listen

Ingenieurinformatik Teil 1, Wintersemester 2026/27

David Straub

Warm-up: Was gibt der Code aus?

Aufschreiben, dann ausführen:

```
def halbiere(x):  
    return x / 2  
  
def zeige(x):  
    print(x)  
  
a = halbiere(10)  
print(a + 1)  
  
b = zeige(10)  
print(b)
```

Heute lernen Sie

- Viele Werte in **einer** Variable: Listen
- Zugreifen, erweitern, durchlaufen
- Die Kernmuster jeder Datenauswertung: **Summe, Mittelwert, Maximum, Zählen – von Hand**

Listen

Warum Listen?

```
messung_1 = 15.2  
messung_2 = 16.1  
messung_3 = 14.8  
messung_4 = 17.3  
# ... und bei 1000 Messwerten?
```

Gleiche Diagnose wie bei den Schleifen: Kopieren skaliert nicht.

Eine Liste erstellen

```
messungen = [15.2, 16.1, 14.8, 17.3]
```

```
print(messungen)
print(len(messungen))
```

- Eckige Klammern, Werte durch Kommas getrennt
- `len(...)` kennen Sie von Strings – funktioniert genauso

Zugriff über den Index

```
messungen = [15.2, 16.1, 14.8, 17.3]

print(messungen[0])    # erstes Element
print(messungen[2])
print(messungen[-1])   # letztes Element
```

- Die Zählung beginnt bei **0** – wie bei `range` !
- Negative Indizes zählen vom Ende: `[-1]` ist das letzte Element

Elemente anhängen: `append`

```
messungen = []           # leere Liste
messungen.append(15.2)
messungen.append(16.1)
print(messungen)
```

- Neue Schreibweise mit Punkt: `append` gehört zur Liste (eine „Methode“)
- Das Muster *leere Liste + anhängen in der Schleife* kommt gleich groß raus

Ist ein Wert enthalten? `in`

```
messungen = [15.2, 16.1, 14.8]

print(16.1 in messungen)
print(99.0 not in messungen)
```

Mini-Aufgabe (3 min)

```
temperaturen = [18.5, 21.0, 19.8, 23.4, 20.1]
```

1. Geben Sie das **letzte** Element aus (`print`) – mit negativem Index
2. Hängen Sie den Messwert `22.7` an (`append`) und geben Sie die Liste aus (`print`)
3. Prüfen Sie mit `in` , ob `20.1` in der Liste ist

Achtung typischer Fehler

```
werte = [10, 20, 30]
print(werte[3])
```

```
IndexError: list index out of range
```

- Drei Elemente heißt: Indizes 0, 1, 2 – der letzte Index ist `len(werte) - 1`
- Oder einfach `werte[-1]` – das geht nie daneben

Listen durchlaufen

for über eine Liste

```
messungen = [15.2, 16.1, 14.8, 17.3]

for messung in messungen:
    print(messung)
```

Die Schleife besucht **jedes Element** der Reihe nach – kein Index nötig.

Testdaten per Zufall: randint

```
from random import randint

werte = []
for i in range(10):
    werte.append(randint(1, 100))

print(werte)
```

- `randint(1, 100)`: ganze Zufallszahl von 1 bis 100 (beide inklusive)
- *Leere Liste + Schleife + append* – dieses Muster brauchen Sie ständig: Testdaten, Simulationen, Messreihen

Kernmuster 1: Summe und Mittelwert von Hand

```
summe = 0
for wert in werte:
    summe += wert

mittel = summe / len(werte)
print(f"Mittelwert: {mittel:.1f}")
```

- **Akkumulator-Muster:** mit 0 starten, in der Schleife aufaddieren
- Es gibt fertige Funktionen (`sum()` , `max()`) – aber das Muster dahinter ist das Handwerk: es trägt jede Auswertung, für die es **keine** fertige Funktion gibt

Kernmuster 2: Maximum von Hand

```
werte = [4, 12, 7, 15, 9]

maximum = werte[0]
for wert in werte:
    if wert > maximum:
        maximum = wert

print(maximum)
```

- Der **Merker** startet mit dem ersten Element
- Jedes größere Element überschreibt ihn – am Ende bleibt das größte

Trace-Aufgabe (2 min, auf Papier)

werte = [4, 12, 7, 15, 9] – bei welchen Werten ändert sich maximum ?

```
maximum = werte[0]
for wert in werte:
    if wert > maximum:
        maximum = wert
```

wert	wert > maximum ?	maximum danach
4	–	4
12	?	?
7	?	?
15	?	?
9	?	?

Debug-Aufgabe (3 min)

Dieses Programm soll das Maximum finden. Für [8, 14, 22] stimmt es – für [-5, -2, -9] liefert es 0. Warum? Beheben Sie den Fehler:

```
werte = [-5, -2, -9]

maximum = 0
for wert in werte:
    if wert > maximum:
        maximum = wert
```

```
print(maximum)
```

Mini-Aufgabe (3 min)

```
werte = [8, 14, 3, 22, 11, 6]
```

Zählen Sie mit einer Schleife, wie viele Werte **über 10** liegen, und geben Sie die Anzahl aus (`print`).

Tipp: Das ist wieder das Akkumulator-Muster – nur zählen Sie statt zu summieren.

Transfer

Aufgabe: Messreihe auswerten (Denkphase: 5 min, auf Papier)

Schreiben Sie ein Programm:

1. Eine Funktion `mittelwert(werte)`, die den Mittelwert einer Liste zurückgibt (`return`)
2. Hauptprogramm: Liste mit **20 Zufallszahlen** zwischen 1 und 100 füllen (`randint`)
3. Funktion aufrufen und das Ergebnis mit einer Nachkommastelle ausgeben (`print`)

Notieren Sie zuerst: Was gehört in die Funktion, was ins Hauptprogramm? (Die Frage aus Woche 6!)

Zusatz für Schnelle: Auch `maximum(werte)` als Funktion – ohne `max()`, versteht sich.

Gemeinsam live

Wir entwickeln die Lösung jetzt zusammen.

Mini-Check

Ohne Computer:

1. `werte = [5, 8, 2]` – was ist `werte[-1]` ?
2. Maximum-Suche: Womit startet man den Merker – und warum nicht mit `0` ?
3. `len([10, 20, 30])` ?
4. Was macht `werte.append(7)` ?

Bis nächste Woche!

Nächste Woche: **Strings** – außerdem eine kurze **Mini-Probeklausur** (15 min, zählt nicht, zeigt Ihnen, wo Sie stehen)

- Üben: KI-Quizze auf OneTutor: <https://hm.onetutor.ai/>
- Fragen: jederzeit im Matrix-Chat