

Python Grundlagen

Listen und Datenstrukturen



Google Colab

Lernziele

Am Ende dieser Lektion können Sie:

- ✓ Listen erstellen und verstehen
- ✓ Auf Elemente zugreifen (Index & Slicing)
- ✓ Listen verändern und Operationen durchführen

Was ist eine Liste?

Liste = Geordnete Sammlung von Elementen

- ▶ Veränderbar (mutable)
- ▶ Kann verschiedene Datentypen enthalten
- ▶ Elemente in eckigen Klammern []

```
1 meine_liste = [1, 2, 3, 'Schule']
```

Listen erstellen

```
1 # Leere Liste  
2 leer = []  
  
3  
4 # Gemischte Datentypen  
5 gemischt = [1, "Hallo", 3.14, True]  
  
6  
7 # Verschachtelte Listen  
8 matrix = [[1, 2], [3, 4]]
```

type(meine_liste) gibt <class 'list'> zurück

Zugriff über Index

```
1 meine_liste = [1, 2, 3, 'Schule']  
2  
3 print(meine_liste[0])      # 1 (erstes Element)  
4 print(meine_liste[3])      # 'Schule'  
5 print(meine_liste[-1])     # 'Schule' (letztes Element)
```

Der Index beginnt bei 0!
[1] ist das **zweite** Element.

Slicing: Teile extrahieren

Syntax: liste[start:stop:step]

```
1 liste = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2
3 print(liste[2:6])          # [3, 4, 5, 6]
4 print(liste[1:8:3])        # [2, 5, 8] (jede 3. Zahl)
5 print(liste[-3:-1])        # [8, 9] (von hinten)
```

Elemente ändern

Listen sind **veränderbar** (mutable):

```
1 meine_liste = [1, 2, 3, 'Schule']
2
3 meine_liste[2] = 99
4
5 print(meine_liste) # [1, 2, 99, 'Schule']
```

Elemente hinzufügen

```
1 meine_liste = [1, 2, 3]
2
3 # Am Ende anhängen
4 meine_liste.append(6)
5
6 # An bestimmter Position einfügen
7 meine_liste.insert(1, 10) # Index 1
8
9 print(meine_liste) # [1, 10, 2, 3, 6]
```

Elemente entfernen

```
1 # Nach Wert entfernen (erstes Vorkommen)
2 meine_liste.remove(2)
3
4 # Nach Index entfernen + Rückgabe
5 letztes = meine_liste.pop()          # letztes Element
6 erstes = meine_liste.pop(0)          # Index 0
```

Häufige Fehler

```
meine_liste.remove(5)    # ValueError: x not in list
```

```
meine_liste.pop(10)      # IndexError: pop index out of  
                        range
```

Vorher prüfen: if 5 in meine_liste:
oder: try: ... except ValueError:

Wichtige Methoden

```
1 liste = [3, 5, 1, 0, 3, 5]
2
3 len(liste)                  # 6 (Länge)
4 liste.sort()                 # Sortiert: [0, 1, 3, 3, 5, 5]
5 liste.reverse()               # Umkehren: [5, 5, 3, 3, 1, 0]
6 liste.index(1)                # 4 (Position des Werts 1)
7 liste.count(3)                # 2 (wie oft kommt 3 vor?)
```

Listen verbinden

```
1 liste1 = [1, 2, 3]
2 liste2 = [4, 5, 6]
3
4 # Mit +
5 neue_liste = liste1 + liste2
6 # [1, 2, 3, 4, 5, 6]
7
8 # Mit extend()
9 liste1.extend(liste2)
```

Mini-Übung 1

Erstelle eine Liste mit den Zahlen 10, 20, 30, 40, 50:

1. Gib das erste und letzte Element aus
2. Füge 60 am Ende hinzu
3. Ändere das Element mit Wert 30 zu 99

Lösung: Mini-Übung 1

```
zahlen = [10, 20, 30, 40, 50]

print(zahlen[0])          # 10
print(zahlen[-1])         # 50

zahlen.append(60)
zahlen[2] = 99            # [10, 20, 99, 40, 50, 60]
```

Mini-Übung 2

Gegeben: liste = [3, 5, 1, 0, 3, 5]

1. Sortiere die Liste
2. Finde den Index der Zahl 1
3. Zähle, wie oft die Zahl 3 vorkommt

Lösung: Mini-Übung 2

```
liste = [3, 5, 1, 0, 3, 5]

liste.sort()          # [0, 1, 3, 3, 5, 5]
print(liste.index(1)) # 1
print(liste.count(3)) # 2
```

Übung: Listen-Operationen

1. Erstelle zwei Listen: `a = [1, 2, 3]` und `b = [4, 5, 6]`
2. Kombiniere sie zu einer Liste `c`
3. Extrahiere mit Slicing die mittleren 4 Elemente
4. Entferne das letzte Element aus `c`
5. Überprüfe mit `in`, ob 3 in `c` enthalten ist

Lösung

```
a = [1, 2, 3]
b = [4, 5, 6]
c = a + b                      # [1, 2, 3, 4, 5, 6]
mitte = c[1:5]                   # [2, 3, 4, 5]
c.pop()                          # [1, 2, 3, 4, 5]
print(3 in c)                   # True
```

Zusammenfassung

- ▶ Listen: [] - veränderbar, geordnet
- ▶ Index: [0], [-1], [start:stop]
- ▶ Methoden: append(), insert(), remove(),
pop()
- ▶ Funktionen: len(), sort(), index(), count()

Hausaufgaben

1. Erstelle eine Einkaufsliste mit 5 Produkten
2. Schreibe ein Programm, das die Liste sortiert und das 2. Produkt entfernt
3. Erstelle ein 2D-Array (Matrix) und greife auf Elemente zu