

17 Datentyp Liste

17.1 Beschreibung

- eine Liste ist eine Sequenz von Elementen
- die Elemente der Liste sind änderbar
- eine Liste kann unterschiedliche Datentypen enthalten
- eine Liste kann wiederum eine Liste enthalten
- eine Liste wird durch die Zuweisung von Elementen innerhalb eckiger Klammern angelegt
- mit Listen-Operatoren und Listen-Funktionen lassen sich Listen ändern und untersuchen
- es lassen sich ein- und mehrdimensionale Felder (Arrays) abbilden
- gleiche Werte können mehrfach vorkommen

- Beispiel:

```
# Liste anlegen
zahlen = [1, 5, 7, 9, 22, 35, 1]
# Liste ausgeben
print(zahlen)
```

```
[1, 5, 7, 9, 22, 35, 1]
```

```
# 1. Element der Liste ausgeben (index fängt bei 0 an)
print(zahlen[0])
```

```
1
```

```
# 2. bis 4. Element der Liste ausgeben (index = 1:4)
print(zahlen[1:4])
```

```
[5, 7, 9]
```

17.2 Listen-Operatoren

- Verkettung von Listen mit `+`
- Vervielfachen von Listen mit `*`
- Zugriff auf ein Element per Index (Zählweise beginnt bei 0) mit `[i]`
- Zugriff auf einen Bereich von Elementen per Index (Zählweise beginnt bei 0) mit `[a:b]`
- Prüfen, ob ein Element in der Liste enthalten ist mit `in`
- Prüfen, ob ein Element in der Liste NICHT enthalten ist mit `not in`
- Element an einem angegebenen Index löschen mit `del liste[i]`
- Bereich von Elementen per Index löschen mit `del liste[a:b]`

- Beispiele:

```
# Liste verketteten
list_t1 = ["Das", "ist"]
list_t2 = ["eine", "Addition"]
list_gesamt = list_t1 + list_t2
print(list_gesamt)
```

```
['Das', 'ist', 'eine', 'Addition']
```

```
# Element in einer Liste verändern
list_2 = ["das", "ist", "eine", "Liste"]
print(list_2)
```

```
['das', 'ist', 'eine', 'Liste']
```

```
list_2[0] = "Das"
print(list_2)
```

```
['Das', 'ist', 'eine', 'Liste']
```

```
# Zugriff auf einen Bereich von Elementen in einer Liste (mit Index)
print(list_2[2:4])
```

```
['eine', 'Liste']
```

```
# Vervielfachen von Listen
list_3 = ["Python", "ist", "eine", "Programmiersprache"]
print(list_3)
```

```
['Python', 'ist', 'eine', 'Programmiersprache']
```

```
print(2*list_3)
```

```
['Python', 'ist', 'eine', 'Programmiersprache', 'Python', 'ist', 'eine', 'Programmiersprache']
```

Aufgabe

- Füge zwei vorher festgelegte Listen zu einer zusammen und gib diese aus
Beispiel: [3,8,9,2] zusammengefügt mit [4,6] ergibt [3,8,9,2,4,6]
- Trenne eine Liste an einem Index in zwei Teillisten.
Hinweis: Der Index beginnt bei 0. Getrennt wird vor dem Eintrag an der Indexstelle. Soll eine Liste an Index 0 getrennt werden, dann ist die erste Liste leer.
Beispiel: [1,3,5,8,9,3] getrennt an Index 3 ergibt [1,3,5] [8,9,3]