

List

Eigenschaften

- Menge von Werten verschiedener Datentypen
- Werte können mehrfach vorkommen
- Die Werte sind geordnet und indexiert
- Die List kann verändert werden

Eine List erzeugen

- Direkte Zuweisung
- Leere Liste erzeugen
- Aus anderen Collections (Mengentypen) erzeugen
- Mit Hilfe eines Generators erzeugen

Direkte Zuweisung

```
list1 = ["Hallo", 3, 1.25, "Welt"]
```

Leere List erzeugen

```
list1 = list()  
list2 = []
```

Aus anderen Collections erzeugen

```
set1 = {"Hallo", 3, 1.25, "Welt"}  
list2 = ["Hallo", 3, "Jahr 2020", 5]  
list1 = list(set1)  
list3 = list2.copy()  
list3 = list(list2)  
list3 = list1 + list2
```

list3 = list2 erzeugt keine neue List
list3 ist nur ein Verweis auf list2!

Mit Hilfe eines Generators erzeugen

```
list1 = list(range(1, 5))
```

Lists untersuchen und vergleichen

- Existiert ein Element
- Elemente Auflisten
- Anzahl Elemente
- Einzelnes Element auslesen
- Anzahl eines Wertes
- Index eines Elementes auslesen
- Teilbereiche auslesen

Hier werden alle Funktionen unterstützt, die wir auch vom Tuple kennen. Die Indexierung folgt ebenfalls denselben Grundregeln.

Existiert ein Element

```
if "Hallo" in list1:
```

Elemente auflisten

```
for x in list1:  
    print(x)
```

Anzahl Elemente

```
anzahl = len(list1)
```

Einzelnes Element auslesen

```
wert = list1[2]
```

Anzahl eines Wertes

```
anz = list1.count("Hallo")
```

Index eines Wertes auslesen

```
idx = list1.index("Hallo") # 1. Vorkommen
```

Der Wert muss existieren, sonst erscheint eine Fehlermeldung

Da die Werte indexiert sind, kann über den Index auf einzelne Elemente und ganze Teilbereiche der List zugegriffen werden.

Das erste und das letzte Element

```
erstes = list1[0]  
letztes = list1[-1]
```

Der Anfang und das Ende

```
anfang = list1[:3] # Element 0 bis 2  
ende = list1[4:]   # Element 4 bis Ende
```

Ein Bereich aus der Mitte

```
bereich = list1[2:5]    # Elemente 2 bis 4  
bereich = list1[-3:-1]  # Drittletztes  
                        Element bis  
                        zweitletztes  
                        Element
```

Diese Bereiche sind keine neuen Listen. Es sind nur Verweise auf den Bereich innerhalb der ursprünglichen List!

```
bereich = list1[2:5].copy()  
bereich = list(list1[2:5])
```

Hier ist ***bereich*** eine eigenständige Liste.

Eine List bearbeiten

Die List kennt vielfältige Bearbeitungsmöglichkeiten.

Die List vollständig löschen

```
del list1
```

Alle Elemente einer List löschen

```
list1.clear()
```

Einzelne Elemente löschen

```
list1.remove("Hallo") # muss existieren,  
                      # löscht erstes  
                      # Vorkommen  
list1.pop()          # löscht letztes Element  
list1.pop(2)          # löscht Element mit Index 2
```

Elemente hinzufügen

```
list1.append("Hi")    # fügt "Hi" am  
                     # Ende hinzu  
  
list1.insert(2, "neu") # fügt "neu" in  
                     # Indexposition 2  
                     # ein  
  
list1.append(list2)   # fügt list2 als  
                     # verschachteltes  
                     # Einzelelement ans  
                     # Ende von List1  
                     # an
```

```
list1.extend([3, "Hallo", 3, "Hallo", "Welt"])  
# fügt die angegebenen Elemente an das  
Ende von list1 an
```

Sortierung

```
list1 = [3, "Hallo", 3, "Hallo", "Welt", 2, 4]  
list1.reverse()
```

```
>>> [4, 2, 'Welt', 'Hallo', 3, 'Hallo', 3]
```

```
list1 = [3, 5, 2, 4]  
list1.sort()
```

```
>>> [2, 3, 4, 5]
```

```
list1 = ["Kirsche", "Banane", "Apfel"]  
list1.sort()
```

```
>>> ['Apfel', 'Banane', 'Kirsche']
```

.sort() funktioniert nur bei gleichen Typen