

💡 Listen

In vielen Fällen ist es sinnvoll oder sogar notwendig, mehrere Dinge in einer Variablen zu speichern. Bei der Programmierung mit Python verwendet man dazu **Listen**.

Eine Liste enthält Elemente in einer festen Reihenfolge. Elemente der Liste können dabei auch mehrfach vorkommen und unterschiedliche Datentypen haben.

Zum Anlegen der Liste verwendet man eckige Klammern, in denen die Elemente der Liste durch Komma getrennt aufgelistet werden.

Auf die Elemente der Liste kann man zugreifen, indem man hinter den Namen der Variablen in eckigen Klammern die Stelle des Elements in der Liste angibt. Die Stelle des Elements nennt man auch den **Index** des Elements. **Achtung: Die Nummerierung beginnt bei 0.**

Die Länge der Liste kann man mithilfe von `len(liste)` abrufen.

Der folgende Code legt zum Beispiel eine Liste mit den ersten 6 Zahlen der Fibonacci-Folge an, speichert die 4. Fibonacci-Zahl in einer neuen Variablen und gibt dann die Länge der Liste aus:

```
1 liste = [1, 1, 2, 3, 5, 8]
2 zahl4 = liste[3] #Wert ist 3
3 print(len(liste)) #Ausgabe: 6
```

👤 Aufgabe 1: Ausgabe eines Listeneintrags

Legen Sie zwei Listen mit jeweils 10 Elementen an. Die erste Liste soll nur Integer beinhalten, die zweite Liste nur Strings.

Lassen Sie den Nutzer nun eine Zahl n eingeben. Geben Sie das n -te Wort in der Liste so häufig aus, wie in der anderen Liste an Stelle n angegeben.

Beispiel:

```
1 woerter = ["Wort1", "Wort2", "Wort3"]
2 zahlen = [2, 4, 6]
3 # Ausgabe bei Nutzereingabe "2": Wort2Wort2Wort2Wort2
```

Wichtige Listenfunktionen

Python bietet viele Funktionen, um mit Listen zu arbeiten. Die wichtigsten sind die folgenden:

- `x in list`: überprüft, ob `x` in der Liste `list` enthalten ist. Das ist vor allem für Verzweigungen wichtig!
- `list.append(x)`: Hängt `x` an die Liste `list` *hinten* an.
- `list.insert(i, x)`: Fügt `x` an die *i*-te Stelle der Liste `list` ein und verschiebt alle nachfolgenden Elemente nach hinten.
- `list.remove(x)`: Entfernt den ersten Eintrag von `x` aus der Liste `list`.
- `list.index(x)`: Findet den Index, an der `x` das erste Mal in der Liste `list` vorkommt.

Aufgabe 2: Elemente tauschen

Lassen Sie den Nutzer 3 Zahlen eingeben, speichern Sie diese in einer Liste und geben Sie diese aus.

Tauschen Sie nun die 1. und die 3. Zahl innerhalb der Liste und geben Sie die veränderte Liste aus.

Aufgabe 3: Einkaufsliste

Legen Sie eine Einkaufsliste mit 5 Einträgen an. Fragen Sie den Nutzer nun nach einer Eingabe:

- Wenn die Eingabe schon auf der Liste steht, löschen Sie den Eintrag.
- Wenn die Eingabe nicht auf der Liste steht, fügen Sie sie hinzu.

Geben Sie zum Schluss die Einkaufsliste aus.

Bonusaufgabe 4: Weitere Funktionen und Slicing

Legen Sie eine Liste `list` mit mindestens 4 Elementen an. Testen und beschreiben Sie die folgenden Listenfunktionen:

- `list.sort()`

- `list.reverse()`

- `list.pop(1)`

- `list[-2]`

- `list[:3]`

- `list[1:]`

- `list[2:3]`
