

 Ziel:

Du lernst den Unterschied zwischen **Listen** und **Tupeln** kennen. Beide speichern mehrere Werte, aber sie verhalten sich unterschiedlich.

Beispiel 1: Liste

```
# Eine Liste ist veränderbar
farben = ["rot", "blau", "grün"]

# Wir geben die Liste aus
print(farben)

# Wir ändern das erste Element(0)
farben[0] = "gelb"
print(farben)
```

✓ **Erklärung:**

- Listen verwendet man, wenn man etwas ändern will.
- farben[0] ist das erste Element. Mit = können wir es ändern.
- print(farben) zeigt die aktuelle Liste an.

 Aufgabe 1:

1. Erstelle eine Liste mit deinen 3 Lieblingsobstsorten.
2. Ändere das zweite Element der Liste.
3. Gib die Liste vorher und nachher mit print() aus.

✓ **Lösung 1:**

```
# Erstelle eine Liste mit Lieblingsobstsorten
obst = ["Apfel", "Banane", "Kirsche"]
print("Vorher:", obst)

# Ändere das zweite Element
obst[1] = "Orange"
print("Nachher:", obst)
```

Beispiel 2: Tupel

```
# Ein Tupel ist unveränderbar
person = ("Anna", 25)
```

```
# Wir geben das Tupel aus
print(person)
```

✓ Erklärung:

- Tupel verwendet man, wenn sich die Werte nicht ändern sollen.
- Sie sind sicherer, weil niemand sie aus Versehen verändert.
- Typisches Beispiel: Name + Geburtsjahr, Koordinaten, feste Daten.

✚ Aufgabe 2:

1. Erstelle ein Tupel mit deinem Namen und deinem Alter.
2. Gib das Tupel mit print() aus.
3. Was passiert, wenn du versuchst, das Alter im Tupel zu ändern?

✓ Lösung 2:

```
# Erstelle ein Tupel mit Name und Alter
person = ("Max", 30)
print(person)
```

```
# Versuch, das Alter zu ändern (führt zu Fehler)
# person[1] = 31 # TypeError: 'tuple' object does not support item assignment
```

💡 Zusammenfassung:

Merkmal	Liste	Tupel	Beispiel
Klammern	[]	()	["rot", "blau"] / ("Anna", 25)
Veränderbar	☒ Ja	☒ Nein	
Verwendung	Wenn Daten sich ändern	Wenn Daten gleich bleiben	