

# Referenzen - Übung

---

Richard Müller, Tom Felber

25. November 2021

Python-Kurs

1. Übung 1 - Referenzen

2. Übung 2 - Vererbung

# Übung 1 - Referenzen

---

# Referenzen - Übung 1.1

```
1 a = []  
2 b = [1]  
3 a.append(b)  
4 a.append(b)  
5 b.append(5)
```

a = ?

b = ?

# Referenzen - Übung 1.1

```
1 a = []  
2 b = [1]  
3 a.append(b)  
4 a.append(b)  
5 b.append(5)
```

a = [ [1, 5], [1, 5] ]

b = [1, 5]

## Referenzen - Übung 1.2

```
1 def f(liste):  
2     liste.remove(1)  
3  
4 a = [1, 2, 3]  
5 f(a)
```

a = ?

## Referenzen - Übung 1.2

```
1 def f(liste):  
2     liste.remove(1)  
3  
4 a = [1, 2, 3]  
5 f(a)
```

a = [2, 3]

## Referenzen - Übung 1.3

```
1 def f(liste):  
2     ???
```

```
1 a = [4, 5, 6]  
2 b = f(a)
```

**Start:**

`a = [4, 5, 6]`

**Ende:**

`a = [4, 5, 6]`

`b = [4, 6]`



## Referenzen - Übung 1.3

```
1 def f(liste):  
2     l = liste[:]  
3     l.remove(5)  
4     return l  
5  
6 a = [4, 5, 6]  
7 b = f(a)
```

**Start:**

a = [4, 5, 6]

**Ende:**

a = [4, 5, 6]

b = [4, 6]

## Übung 2 - Vererbung

---

# Klassen - Übung 2

1. Definiere eine Klasse `Fahrzeug` mit folgenden Eigenschaften:

- Attribute: `speed` , `preis`
- Methode: `hupen()` soll 'Huuup' ausgeben
- Methode: `info()` soll einen String mit Preis und Geschwindigkeit zurückgeben (!nicht printen sondern zurückgeben)

Überprüfe, ob die Klasse funktioniert, indem du mehrere Instanzen in einer Liste erstellst und alle `hupen()` lässt:

```
1 fahrzeuge = [Fahrzeug(1000, 10), Fahrzeug(20000, 100)]  
2  
3 for fahrzeug in fahrzeuge:  
4     print(fahrzeug.info())  
5     fahrzeug.hupen()
```

2. Füge eine weitere Methode `fahren(dauer)` hinzu, die eine `Dauer` entgegen nimmt. Sie soll die Distanz zurückgeben, die in der angegebenen Zeit zurückgelegt wurde. ( $dauer * speed$ )

Überprüfe, ob die Klasse funktioniert, indem du mehrere Instanzen in einer Liste erstellst und alle `fahren()` lässt:

```
1 fahrzeuge = [Fahrzeug(1000, 10), Fahrzeug(20000, 100)]
2
3 for fahrzeug in fahrzeuge:
4     print(fahrzeug.info())
5     fahrzeug.hupen()
6     fahrzeug.fahren(100)
```

3. Definiere die Klasse `PKW` und `LKW`. Beide sollen von `Fahrzeug` erben. Der LKW soll außerdem `hupen` überschreiben (anderes Geräusch ausgeben).
4. Erweiterung der Klassen.

## PKW

- `__init__` überschreiben, alle PKWs sollen 30.000 € kosten und 180 km/h schnell sein
- zusätzliche Attribute: `personen_zahl`
- `info()` Methode überschreiben: soll `personen_zahl` mit ausgeben

## LKW

- `__init__` überschreiben, alle LKWs sollen 200.000 € kosten und 80 km/h schnell sein
- zusätzliche Attribute: `laderaum`
- `info()` Methode überschreiben: soll `laderaum` mit ausgeben

**Zusatz:** Definiere eine **Panzer** Klasse, die auch von Fahrzeug erbt. Panzer sollen 6.000.000 € kosten, 70 km/h schnell fahren und eine Anzahl von Munition haben. Außerdem soll ein Panzer **schießen** können, wobei Munition verbraucht wird. Ist keine Munition mehr vorhanden, so kann er keinen Schuss mehr abgeben. Beim Schießen soll zudem 'Feuer!' ausgegeben werden, wenn der Schuss erfolgt.