

# Übungen zu Referenzen und Gültigkeitsbereichen

---

## Aufgabe 1

Gib die Werte an, die a und b nach dem Aufruf von shadow referenzieren, und erkläre warum:

```
>>> def shadow(a):  
    b = a  
  
>>> a = 55  
>>> b = 66  
>>> shadow(30)  
>>> a  
  
>>> b
```

## Aufgabe 2

Gleiche Aufgabe wie in Aufgabe 1, aber mit folgender Funktion:

```
>>> def shadow(c):  
    global b  
    b = a
```

## Aufgabe 3

Zeichne drei Diagramme, welche zeigen, welche Namen auf welche Werte verweisen a) vor dem Funktionsaufruf, b) unmittelbar vor dem Ende des Funktionsaufrufs und c) nach dem Funktionsaufruf von step.

```
def step(pos, v):  
    pos[0] = pos[0] + v[0]  
    pos[1] = pos[1] + v[1]  
  
ort = [5,2]  
geschwindigkeit = [3,4]  
step(ort, geschwindigkeit)
```

## Aufgabe 4

Zeichne drei Diagramme, welche zeigen, welche Namen auf welche Werte verweisen a) vor dem Funktionsaufruf, b) unmittelbar vor dem Ende des Funktionsaufrufs und c) nach dem Funktionsaufruf von step.

```
def step(pos, v):  
    x = pos[0] + v[0]  
    y = pos[1] + v[1]  
    return [ x, y ]  
  
ort = [5,2]  
geschwindigkeit = [3,4]  
neu = step(ort, geschwindigkeit)
```

## Aufgabe 5

Was ist der grundlegende Unterschied zwischen den Programmen in Aufgabe 3 und in Aufgabe 4?

## Aufgabe 6

- Unterstreiche *alle* lokalen Namen blau, alle globalen rot und alle Funktionsaufrufe grün
- Mache zwei Listen *aller* Namen, die in `vorhandeneDinge` und in `sichtbareDinge` an Werte gebunden sind (ohne Angabe der Werte).

```
haus = [ 'Haus', 'Du stehst in einem alten unheimlichen Haus.' ]
schwert = [ 'Schwert', 'Ein rostiges Schwert' ]
truhe = [ 'Truhe', 'Eine uralte Truhe aus Eichenholz' ]
objekte = [ schwert, truhe ]
verschlossen = [ truhe ]
ist_in = [ (schwert, truhe), (truhe, haus) ]
aktuellerRaum = haus ❶
```

```
def vorhandeneDinge(wo):
    hier = []
    for ding, container in ist_in:
        if container == wo:
            hier.append(ding)
    return hier
```

```
def sichtbareDinge(wo):
    sichtbar = vorhandeneDinge(wo) ❷
    for ding, container in ist_in:
        if container in sichtbar and container not in verschlossen:
            sichtbar.extend(sichtbareDinge(container))
    return sichtbar
```

```
print sichtbareDinge(aktuellerRaum)
```

## Aufgabe 7

Was haben die Namen `wo`, `ding` und `container` in der Funktion `vorhandeneDinge` und in der Funktion `sichtbareDinge` miteinander zu tun?

## Aufgabe 8

Zeichne ein Diagramm, das aufzeigt, welche Werte die globalen Namen im Programm aus Aufgabe 6 unmittelbar nach der Ausführung der mit ❶ bezeichneten Zeile referenzieren.

## Aufgabe 9

Mache für das Programm in Aufgabe 6 ein Diagramm, in welchem du einzeichnest, welche lokalen und globalen Namen welche Werte referenzieren unmittelbar nach der Ausführung der mit ❷ bezeichneten Zeile.