### *Zum Sichern des Besten call-by-reference nutzen*

Nach der Bearbeitung in `fuelle_ein` ist `container` nicht in der aufrufenden Stufe von `fuelle` geändert!

```
def fuelle_ein(stueck, container):
    fuellung=[]+container[1] # echte Kopie erzeugen!
    fuellung.append(stueck)
    return [container[0], fuellung]

# Prädikat zum Testen, ob der Container voll ist.
def ist_voll(container):
    return container[0]==sum(container[1])

def wird_zu_voll(stueck, container):
    return container[0]<stueck+sum(container[1])

def inhalt(container):
    return sum(container[1])

def fuelle(stuecke, container, bester):
    ##input(str((stuecke,container,bester))) ## zum Verfolgen des Ablaufs
    print(stuecke,container,bester)
    if ist_voll(container):
        return container
    if len(stuecke)==0:
        return bester
    if wird_zu_voll(stuecke[0], container):
        return False
    ## Verwaltung fuer die Bestensuche:
    neu=fuelle_ein(stuecke[0], container)
    if inhalt(neu)>inhalt(bester):
        bester=neu
    ## weiter die eigentliche Suchfunktion
    ergebnis=fuelle(stuecke[1:],neu, bester)
    if ergebnis:
        return ergebnis
    else:
        return fuelle(stuecke[1:], container, bester)
```


Nach der Bearbeitung des Aufrufs von `fuelle` ist `bester` auch in der aufrufenden Stufe von `fuelle` geändert!