

Abschluss Funktionen (Musterlösung)

Nimm dir die Zeit, um die Aufgaben sorgfältig zu bearbeiten. :-) Viel Erfolg!

Ein funktionaler Online-Shop

Die Mathemagierin will ihren Online-Shop auf Funktionen umrüsten. Es wartet also wieder einiges an Arbeit auf dich.

a.) Schreibe eine Funktion, die den Gesamtpreis der Produkte im Warenkorb berechnet!

Vervollständige die Funktion `list_sum()`, der als Parameter eine Liste mit den Preisen übergeben wird. Die Funktion soll dann die Summe der Zahlen aus der Liste ausgeben.

In [1]:

```
cart_prices = [20, 3.5, 6.49, 8.99, 9.99, 14.98]

def list_sum(l):
    total = 0
    for i in l:
        total = total + i
    print(total)
    # alternativ einfach: print(sum(l))

list_sum(cart_prices)
```

63.95

Folgende Ausgabe wird erwartet: 63.95

b.) Schreibe eine Funktion, die für einen Artikel eine Preis-Tabelle erstellt!

Nun wünscht sich die Mathmagierin eine Funktion, der sie einen Artikelnamen und den Verkaufspreis übergeben kann. Daraus soll die Funktion eine Liste erstellen, in der die Preise von einem, zwei, drei,... bis zehn Einheiten des Artikels stehen. Genauer soll jedes Element in der Liste so aussehen: "Anzahl x Artikel: Preis".

Du wunderst dich nur kurz über die Ansprüche der Mathemagierin.

In [1]:

```
def prices_list(name, price):
    l = []
    for i in range(1, 11):
        l.append(str(i) + " x " + name + ": " + str(price * i))
    return l

print(prices_list("Wunderkeks", 0.79))
```

['1 x Wunderkeks: 0.79', '2 x Wunderkeks: 1.58', '3 x Wunderkeks: 2.37', '4 x Wunderkeks: 3.16', '5 x Wunderkeks: 3.95', '6 x Wunderkeks: 4.74', '7 x Wunderkeks: 5.53', '8 x Wunderkeks: 6.32', '9 x Wunderkeks: 7.11', '10 x Wunderkeks: 7.9']

Folgende Ausgabe wird erwartet (muss nicht farbig sein):

```
['1 x Wunderkeks: 0.79', '2 x Wunderkeks: 1.58', '3 x Wunderkeks: 2.37',
'4 x Wunderkeks: 3.16', '5 x Wunderkeks: 3.95', '6 x Wunderkeks: 4.74',
'7 x Wunderkeks: 5.53', '8 x Wunderkeks: 6.32', '9 x Wunderkeks: 7.11',
```

```
'10 x Wunderkeks: 7.9']
```

c.) Schreibe eine Funktion, die die Listen mit den Artikeln auffüllt!

Von nun an soll auch eine Funktion die Waren in die virtuellen Regale einräumen, d. h. an die erste, noch leere Position in der Liste *shelf* packen. Als Parameter soll der Funktion `add_shelf()` der einzusortierende Artikel übergeben werden. Die Funktion aktualisiert dann die Liste `shelf`, und der neue Artikel wurde in das erste leere Regalfach eingeräumt.

In [5]:

```
shelf = ["Zaubersäge", "leer", "Wunderkekse", "Trickkarten", "leer"]

def add_shelf(article):
    for i in range(0, len(shelf)):
        if shelf[i] == "leer":
            shelf[i] = article
            break

add_shelf("Rubik's Cube")
print(shelf)
```

```
['Zaubersäge', "Rubik's Cube", 'Wunderkekse', 'Trickkarten', 'leer']
```

Folgende Ausgabe wird erwartet (braucht nicht farbig zu sein):

```
['Zaubersäge', "Rubik's Cube", 'Wunderkekse', 'Trickkarten', 'leer']
```

Gute Arbeit :-)