

## Arbeitsblatt: Rekursion II

Falls Sie die Aufgaben einfach finden, implementieren Sie die folgenden Funktionen komplett, d.h. überprüfen Sie auch die übergebenen Parameter ob sie überhaupt sinnvoll sind. Sie können mit `error` „eine Fehlermeldung“ eine Meldung auf der Konsole ausgeben und das Programm abbrechen.

### Aufgabe 1

Implementieren Sie eine Funktion welche die Länge einer Liste bestimmt:

```
len :: [a] -> Int
```

### Aufgabe 2

Implementieren Sie eine Funktion, welche überprüft ob in einer Liste von `Bool` alle Werte `True` sind:

```
allTrue :: [Bool] -> Bool
```

Hinweis: `allTrue []` soll auf `True` evaluieren.

### Aufgabe 3

Implementieren Sie eine Funktion, welche aus einer List eine Teilliste extrahiert, indem dessen Startposition und Länge gegeben ist:

```
sublist :: Int -> Int -> [a] -> [a]
```

```
*Main> sublist 3 7 "Hello World"  
"lo Worl"
```

Wir meinen nicht diese (zugegebenermassen richtige und effektive) Lösung:

```
sublist beg len xs = take len (drop beg xs)
```

Geben Sie eine rekursive Lösung ohne Verwendung von Standardfunktionen!

### Aufgabe 4

Implementieren Sie eine Funktion, welche eine Liste an eine andere Liste anhängt:

```
(++) :: [a] -> [a] -> [a]
```