

## Funktionale Abstraktion

- Funktionen sind zentrale Bausteine
- Programmdesign:
  - Zerlegung der Hauptfunktion in kleinere Funktionen
  - Funktionen rufen andere, Funktionen auf-> "Funktionen höherer Ordnung"
- Funktionen unterscheiden sich durch ihre Parameter, welche wiederum Parameter sind.

## Deklaratives Programmieren

- Funktionale Sprachen = Oberfamilie der deklarativen Sprache
- Compiler wird erzählt, "was" berechnet wird statt wie->Keine Befehle, sondern nur Formeln
- kein zeitlicher Ablauf und keine Objektidentität, Computerspeicher also "egal"

## Referentielle Transparenz

- Jeder Aufruf einer Funktion kann ersetzt werden, durch das Ergebnis der Funktion
  - selbe Parameterwerte liefern selbes Ergebnis
  - keine void Methoden, Funktionen haben immer ein Rückgabewert => keine Seiteneffekte
  - -> in [Racket Einstieg](#) gibt es nur Konstanten
-