






**Präsenzübungen zur Vorlesung**  
**Deklarative Programmierung: Sommersemester 2018**  
Nr. 1





---

**Aufgabe 1.1:** Ausdrücke & Fehler



Welche Ausgabe erwarten Sie von den folgenden Ausdrücken von DrRacket?

- a) `(replicate 7 "Wie viele Tage?")` 
- b) `(+ 1337 (number->string 23))` 
- c) `(= (- 1527 27) (* 500 3))` 
- d) `+ 23 42` 
- e) `(string->number "Twelve")` 

**Aufgabe 1.2:** Bedeutung von BSL-Ausdrücken




- a) Diskutieren Sie den Unterschied zwischen *Werten* und *Ausdrücken*. Geben Sie für jeden der beiden Fälle ein geeignetes Beispiel an. 
- b) Berechnen Sie den Wert des folgenden Ausdrucks, indem Sie diesen schrittweise reduzieren:  
`(* (+ 2 2) (+ (* 4 (/ 20 2)) 41))` 
- c) Wann sind zwei Programme äquivalent? Zeigen Sie die Äquivalenz zweier Programme an einem geeigneten Beispiel. 
- d) Unter welcher Voraussetzung kann ein Unterausdruck eines Programms durch einen anderen ersetzt werden, ohne die Semantik des Programms zu verändern? Geben Sie auch hierfür ein geeignetes Beispiel an. 

**Aufgabe 1.3:** Eigene Funktionen





- a) Geben Sie das allgemeine Schema einer Funktionsdefinition in DrRacket an. 
- b) Betrachten Sie das folgende Programm und begründen Sie, ob es sich dabei um ein gültiges BSL-Programm handelt: 

```
(define temperatur-sonne (fahrenheit->celsius 9941))  
(define (fahrenheit->celsius temp) (* (- temp 32) (/ 5 9)))  
temperatur-sonne
```







### Aufgabe 1.4: DRY

- a) Was verstehen Sie unter dem DRY-Prinzip? 
- b) Erläutern Sie den Begriff *magic-numbers*. 
- c) Wie können *magic-numbers* und *Redundanzen* in Programmen vermieden werden? 

### Aufgabe 1.5: Systematischer Programmentwurf



- a) Warum ist es vorteilhaft Teilaufgaben in eigene Funktionen auszulagern? 
- b) Erläutern Sie den Unterschied zwischen „Top-Down“- und „Bottom-Up“-Entwicklung! 
- c) Was verstehen Sie unter dem Begriff „hierarchische Abstraktion“? 
- d) Was ist *Information Hiding*? 

### Aufgabe 1.6: A New Design Recipe

- a) Geben Sie zu jeder der folgenden Maßeinheiten eine passende Datendefinition an:
  - (i) Preis 
  - (ii) Nachname 
  - (iii) Fläche 
  - (iv) Klausurnote 
  - (v) Körpergewicht 
- b) Diese Datendefinitionen wirken sich in keiner Weise auf Programme aus. Warum sollten Sie diese dennoch angeben? 

### Aufgabe 1.7: Wrath of Design Recipe

Schreiben Sie...

- a) eine Signatur für eine Funktion, die Meilen in Kilometer umrechnet. 
- b) eine Signatur und Aufgabenbeschreibung für eine Funktion, die den durchschnittlichen Kraftstoffverbrauch eines Autos berechnet. 
- c) eine Signatur, eine Aufgabenbeschreibung und einen Funktionskopf für eine Funktion, die prüft, ob ein Punkt (gegeben: Koordinaten x und y) sich in einem Kreis vom Radius r um den Ursprung befindet. 