

Übungen Programmieren in Clojure Serie 1

1. Präfix-Notation

Formulieren Sie folgende Ausdrücke in der Präfix-Notation¹ und werten Sie sie in der Repl aus.

- (a) $2 + 2 + 2 + 3$
- (b) $(2 * (3 + 4 - 2*2))$
- (c) $(42 - 3) - 39$
- (d) $42 - (3 - 39)$
- (e) $2 + 1$ (ohne die Zeichen + und - zu verwenden)
- (f) `true and false or true`
- (g) `(1 = 1) or (1 != 1)`

2. Ausdrücke in der Repl

Was ergeben die folgenden Ausdrücke in der Repl, eingegeben in der vorgegebenen Reihenfolge. Bitte überlegen Sie zuerst im Kopf, was das Ergebnis sein muss und probieren Sie erst dann aus.

```
user=>10

user=>(+ 5 3 4)

user=>(- 9 1)

user=>( / 6 2)

user=>( / 7 2)

user=>( / 8 6)

user=>(+ (* 2 4) (- 4 6))

user=>(def a 3)

user=>(def b (+ a 1))

user=>(+ a b (* a b))

user=>(= a b)

user=>(if (and (> b a) (< b (* a b))) b a)

user=>(cond
  (= a 4) 6
  (= b 6) (+ 6 7 a)
  :else 25)
```

¹LISP = „Lots of Irritating Superfluous Parentheses“ (Wikipedia)

3. Datenstrukturen in der Repl

Was ergeben die folgenden Ausdrücke in der Repl? Bitte überlegen Sie zuerst im Kopf, was das Ergebnis sein muss und probieren Sie erst dann aus.

```
user=>[1 1 1]
user=>(1 1 1)
user=>[+ 1 2]
user=>(+ 1 2)
user=>[(+ 1 2) (inc 3)]
user=>(first [1 2 3 4 5])
user=>(rest [1 2 3 4 5])
user=>#{1 1 1}
user=>#{1 2 3}
user=>{1 \a 2 \b 3 \c}
user=>{1 a 2 b 3 c}
user=>{1 \a 2}
```

4. Daten = Code

- Wie kann man die Liste (+ 2 3) als eine Liste von Werten behandeln?
- Und wie kann man eine Liste (als Daten) mit den Elementen +, 2 und 3 als Code ausführen?