

```
;;;;;;;;;;;;;;  
;;  
;; teh amazing lisp workshop  
;;  
;;  
  
;;;;;;;;;;;;;;  
;;  
;; part1: basics  
;;  
  
  
; this is a comment :)  
  
  
  
  
  
  
  
  
  
; ein funktionsaufruf mit argumenten:  
(+ 1 2 3 4)  
  
  
  
  
  
  
  
  
  
; eine liste  
(foo bar baz)  
  
  
  
  
  
  
  
  
  
; wahr  
t  
  
  
  
  
  
  
  
  
  
; falsch = leere liste  
nil  
( )  
  
  
  
  
  
  
  
  
  
; dinge, die zu sich selbst evaluieren:  
t  
nil
```

23

`"hello world"``; if`

```
(if t
  "jawoll"
  "oh nein")
```

```
(+ 1 2 3 (if t
             4
            -6))
```

```
;
; alles was in () steht,
; evaluiert auch zu etwas
;
```

```
; funktionsdefinition
; defun = "define function"
; (defun <name> <argumente>
;   <body>)
;
```

```
(defun sqr (zahl)
  (* zahl zahl))
```

```
(defun addiere-zahlen (zahl1 zahl2)
  (+ zahl1 zahl2))

; strings formatieren (pretty-print)
(format t "hallo ~a" "foo")

; noch eine funktion
(defun report-club-condition (place state)
  (format t "hallo. wie immer ist die ~a ~a."
    place state))

; und noch eine
(defun print-quadrat (zahl)
  (format t "das quadrat von ~a ist ~a.~%"
    zahl
    (* zahl zahl)))

; quote
(1 2 3 4)
'(1 2 3 4)

foo
'foo
''foo

(format t "~a ist eine tolle liste"
  (eins 2 drei 4))

(format t "~a ist eine tolle liste"
  '(eins 2 drei 4))

;;
;; iterieren
;;

; dolist: ueber listen

(dolist (tolle-zahl '(1 2 3 4))
```

```
(print-quadrat tolle-zahl))

(defun stinker (name list-of-smells)
  (dolist (smell list-of-smells)
    (format t "~a stinkt nach ~a.~%"
            name smell))) ;hier ist ein fehler

; dotimes: 0..n
(dotimes (tolle-zahl 10)
  (print-quadrat tolle-zahl))

;;;;;;;;;;;;;;;;;;
;;
;; skillz, die wir jetzt haben:
;; * listen machen
;; * funktionen anwenden
;; * funktionen definieren: defun
;; * text ausgeben mit format
;; * einfache loops: dotimes, dolist
;; * kontrollfluss: if
;;

;;;;;;;;;;;;;;;;;;
;;
;; work work!
;;
;; * REPL zum laufen bekommen
;; * (* 1 2 3) evaluieren
;; * funktion say-hello schreiben, die fuer
;;   (hello "du da")
;;   "nice to see you, du da!" ausgibt.
;; * alle zahlen von 0 bis 23 ausgeben
;; * alle geraden zahlen von 0 bis
;;   42 ausgeben
;; * funktion schreiben, die das fuer
;;   alle zahlen von 0 bis max (argument
;;   der funktion) tut
;;
;; * play :)
```

```
;;  
;; hints:  
;; * (evenp 4) evaluiert zu t  
;;  
;;
```