

Lisp Befehle Zusammenfassung

QUOTE

'<ARG>

QUOTE gibt einen Wert oder Liste

QUOTE ist eine spezielle Form.

QUOTE hat 1 Argument.

Das Argument kann ein beliebiger Ausdruck sein.

Beispiele für QUOTE

'A

'(A B C)

'(FIRST (A B C))

CAR

CAR <LISTE>

CAR holt das erste Element aus einer Liste.

CAR ist eine Listenfunktion.

CAR hat 1 Argument.

Das Argument muß eine Liste sein.

Beispiele für CAR

(CAR '(1 2 3))

(CAR (CAR '((2KG ZUCKER) (3L MILCH) (1KG MEHL))))

Negative Beispiele für CAR

(CAR (A B C))

(CAR A B C)

CDR

(CDR <LISTE>)

CDR entfernt das erste Element aus einer Liste.

CDR ist eine Listenfunktion.

CDR hat 1 Argument.

Das Argument muß eine Liste sein.

Beispiele für CDR

(CDR '(1 2 3))

(CDR 'NIL)

(CDR '(NIL NIL))

(CDR '((1 2 3) (4 5 6)))

(CDR (CDR '((1 2) (3 4) (5 6))))

Negative Beispiele für CDR

(CDR (A B C))

(CDR 'A 'B 'C)

(CDR 'LISTE)

CONS

(CONS <ARG> <LISTE>)

CONS fügt ein Argument in eine Liste ein.

CONS ist eine Listenfunktion.

CONS hat 2 Argumente.

Die Argumente müssen ein beliebiger Ausdruck und eine Liste sein.

Beispiele für CONS

(CONS 'A '(B C D))

(CONS 'A 'NIL)

(CONS 'A (CONS 'B (CONS 'C 'NIL)))

Negative Beispiele für CONS

(CONS 'A (B C D))

(CONS A '(B C D))

'(CONS 'A '(B C D))

APPEND

(APPEND <LISTE> <LISTE...>)

APPEND fügt zwei Listen zu einer Liste zusammen.

APPEND ist eine Listenfunktion.

APPEND hat beliebig viele Argumente.

Die Argumente müssen alle Listen sein.

Beispiele für APPEND

```
(APPEND '(A B C) '(D E F))
```

```
(APPEND '(GLASS OF TEE) '(WITH MILK AND SUGAR))
```

```
(APPEND NIL NIL)
```

```
(APPEND '((A 1) (B 2)) '((C 3) (D 4)))
```

Negative Beispiele für APPEND

```
(APPEND '(A B C) 'D)
```

```
(APPEND 'A '(B C D))
```

```
(APPEND 1 '(2 3))
```

LIST

(LIST <ARG> <ARG...>)

LIST macht aus einer Anzahl von Argumenten ein Liste.

LIST ist eine Listenfunktion.

LIST hat beliebig viele Argumente.

Alle Argumente können beliebige Ausdrücke sein.

Beispiele für LIST

```
(LIST 'A 'B 'C 'D)
```

```
(LIST (LIST 'A 'B) (LIST 'C 'D 'E))
```

```
(LIST 'NIL 'A 'B '(C D NIL) 'NIL)
```

Negative Beispiele für LIST

```
(LIST A B C D)
```

```
'(LIST 'A 'B 'C 'D)
```

```
(LIST (A B C D))
```

REVERSE

(REVERSE <LISTE>)

REVERSE kehrt die Reihenfolge der Elemente einer Liste um.

REVERSE ist eine Listenfunktion.

REVERSE hat 1 Argument.

Das Argument muß eine Liste sein.

Beispiele für REVERSE

```
(REVERSE '(1 2 3 4 5))
```

```
(REVERSE '((VERY BAD) (BAD) (AVERAGE) (GOOD) (VERY GOOD)))
```

```
(REVERSE NIL)
```

Negative Beispiele für REVERSE

```
(REVERSE 'LIVE)
```

```
(REVERSE 1234)
```

```
(REVERSE '(1 2 3) '(4 5 6))
```

LENGTH

(LENGTH <LISTE>)

LENGTH berechnet die Länge einer Liste (Top-level Elemente).

LENGTH ist eine Listenfunktion.

LENGTH hat 1 Argument.

Das Argument muß eine Liste sein.

Beispiele für LENGTH

```
(LENGTH NIL)
```

```
(LENGTH '(1 2 3 4 5))
```

```
(LENGTH '((1 2 3) (4 5 6) (7 8 9)))
```

```
(LENGTH NIL)
```

Negative Beispiele für LENGTH

```
(LENGTH 123)
```

```
(LENGTH 'FOO)
```

```
(LENGTH '(1 2 3) '(4 5 6))
```

DEFUN

(DEFUN <NAME> (<ARG> <ARG...>) <ANWEISUNGEN>)

DEFUN definiert eine Funktion

DEFUN ist eine spezielle Form.

DEFUN hat 3 Argumente.

Die Argumente müssen ein Symbol, eine Liste und ein beliebiger Ausdruck sein.

Beispiele für DEFUN

```
(DEFUN QUAD (S) (* S S))
```

```
(DEFUN MY-SECOND (LI) (FIRST (REST LI)))
```

```
(DEFUN LAST-EL (LI) (FIRST (REVERSE LI)))
```

```
(DEFUN REPLACE-THIRD (EL LI) (CONS (FIRST LI) (CONS (FIRST (REST LI)) (CONS EL (REST (REST (REST LI)))))))
```

```
(DEFUN MY-MEMBER (EL LI) (COND ((ENDP LI) NIL) ((EQUAL (FIRST LI) EL) LI) (T
```

Negative Beispiele für DEFUN

```
(DEFUN A B (LIST A B))
```

```
(DEFUN '(C D) (E F) (* E F))
```

EQUAL

(EQUAL <ARG> <ARG>)

EQUAL testet, ob zwei Argumente gleich sind.

EQUAL ist ein Prädikat.

EQUAL hat 2 Argumente.

Die Argumente müssen ein beliebiger Ausdruck und ein beliebiger Ausdruck sein.

Beispiele für EQUAL

```
(EQUAL 'HANS 'HANS)
```

```
(EQUAL 'HANS (FIRST '(HANS OTTO GERTRUD)))
```

```
(EQUAL 'HANS (REST (REST '(OTTO GERTRUD HANS))))
```

Negative Beispiele für EQUAL

```
(EQUAL HANS)
```

```
(EQUAL 'HANS OTTO)
```

```
(EQUAL 1 1 1 1)
```

ATOM

(ATOM <ARG>)

ATOM testet ob ein Argument ein Atom ist.

ATOM ist ein Prädikat.

ATOM hat 1 Argument.

Das Argument kann ein beliebiger Ausdruck sein.

Beispiele für ATOM

```
(ATOM 'ABC)
```

```
(ATOM '(A B C))
```

```
(ATOM NIL)
```

```
(ATOM (FIRST '(A B C)))
```

Negative Beispiele für ATOM

```
(ATOM 'A 'B 'C)
```

```
(ATOM 32 23)
```

```
(ATOM '(1 2 3) '(4 5 6))
```

LISTP

(LISTP <ARG>)

LISTP testet, ob ein Argument eine Liste oder NIL ist.

LISTP ist ein Prädikat.

LISTP hat 1 Argument.

Das Argument kann ein beliebiger Ausdruck sein.

Beispiele für LISTP

```
(LISTP 'A)
```

```
(LISTP 999)
```

```
(LISTP '(A B C))
```

```
(LISTP '(THIS IS A LIST))
```

Negative Beispiele für LISTP

```
(LISTP '(1 2 3) '(4 5 6))
```

```
(LISTP 'GLASS 'OF 'TEE)
```

NULL

(NULL <ARG>)

NULL testet auf die leere Liste NIL.

NULL ist ein Prädikat.

NULL hat 1 Argument.

Das Argument kann ein beliebiger Ausdruck sein.

Beispiele für NULL

(NULL T)

(NULL NIL)

(NULL (NULL T))

(NULL (FIRST '(NIL T NIL)))

(NULL 1)

Negative Beispiele für NULL

(NULL NULL NULL)

(NULL FIRST)

COND

(COND (<BEDINGUNG> <KONSEQUENZEN>) <WEITERE_FÄLLE...>)

COND führt die Konsequenz aus, wenn die Bedingung wahr ist.

COND ist eine spezielle Form.

COND hat beliebig viele Argumente.

Die Argumente müssen alle Cond-Klauseln, Listen mit einer Bedingung und Konsequenz(en), sein.

Beispiele für COND

(COND ((NUMBERP 3) 'NUMBER) (T 'NO-NUMBER))

(COND ((NOT (NUMBERP 4)) 'NO-NUMBER) ((> 4 10) 'BIG-NUMBER) ((< 4 1) 'SMALL-SMALL) (T 'NUMBER))

(COND ((NOT NIL) '(LIST OR) 'ATOM) (T 'NEVER))

(COND (T 'FALL-1) (T 'FALL-2) (T 'FALL-3))

(COND ((NOT T) 'FALL-1))

(COND)

Negative Beispiele für COND

(COND (NOT NULL) 'ERROR)

(COND ((NOT T) 'ABC) T (LIST 1 2 3))

IF

(IF <BEDINGUNG> <THEN> <ELSE>)

IF wertet je nach Wert der BEDINGUNG entweder WAHR oder FALSCH aus.

IF ist eine spezielle Form.

IF hat 3 Argumente.

Die Argumente müssen ein beliebiger Ausdruck, ein beliebiger Ausdruck und ein beliebiger Ausdruck sein.

Beispiele für IF

(IF T 'TRUE)

(IF T 'TRUE 'FALSE)

(IF NIL 'TRUE 'FALSE)

(IF 'FALSE T NIL)

Negative Beispiele für IF

(IF (NUMBERP 3) 'YES 'NO 'MAYBE)

(IF T 'A 'B 'C)

(IF)