

# Rekursion Arbeitsweise

Demonstration der Arbeitsweise  
einer **rekursiven** Funktion

am Beispiel  
Bestimmung der Länge  
einer Liste

# Rekursion Arbeitsweise

- Die Funktion:

```
(define
  (laenge liste)
  (cond
    ((null? liste)
     0)
    (else
     (+
      1
      (laenge (rest liste))))
  )))
```

# Rekursion Arbeitsweise

Bestimmung der Länge einer Liste

```
(define
  (laenge liste)
  (if
    (null? liste)
    0
    (+ 1 (laenge (rest liste))))
  )
)
```

# Rekursion Arbeitsweise

Funktionsaufruf:

**(laenge '(a b c d e))**



# Rekursion Arbeitsweise

Funktionsaufruf:

**(laenge '(b c d e))**



# Rekursion Arbeitsweise

Funktionsaufruf:

**(laenge '(c d e))**



# Rekursion Arbeitsweise

Funktionsaufruf:

**(laenge '(d e))**



# Rekursion Arbeitsweise

Funktionsaufruf:  
**(laenge '(e))**



# Rekursion Arbeitsweise

Funktionsaufruf:

(laenge '0)  $\rightarrow 0$



# Rekursion Arbeitsweise

Auswertung:

**(laenge '0) → 0**



# Rekursion Arbeitsweise

Auswertung:

$(+ 1 (\text{laenge } '0)) \rightarrow 1$



# Rekursion Arbeitsweise

Auswertung:

**(+ 1 (laenge '(e)))**       $\rightarrow 2$



# Rekursion Arbeitsweise

Auswertung:

`(+ 1 (laenge '(d e)))` → 3



# Rekursion Arbeitsweise

Auswertung:

`(+ 1 (laenge '(c d e)))`  $\rightarrow 4$



# Rekursion Arbeitsweise

Auswertung:

**(+ 1 (laenge '(b c d e)) )** → 5



# Rekursion Arbeitsweise

Funktionsergebnis ausgeben

**(laenge ' (a b c d e))** → 5

liste				
a	b	c	d	e