

# Funktionale Sprachen: LISP

Beispiel: vier Möglichkeiten  $m^n$  zu berechnen

1. in pure-LISP

```
(defun expt-rek-cond (m n)
  (cond ((zerop n) 1)
        (T (cond ((and (integerp n) (> n 0))
                    (* m (expt-rek-cond m (- n 1))))
            ;  $m^n = m * m^{n-1}$ 
            (T "Exponent muss positiver Integer sein"))
        )
  )
)
```

# Funktionale Sprachen: LISP

## 2. in impure LISP mit bedingtem Ausdruck

```
(defun expt-rekursiv (m n)
  (if (zerop n)
      1 ;Abbruchbedingung:  $m^0=1$ 
      (if (and (integerp n) (> n 0))
          (* m (expt-rekursiv m (- n 1))) ;  $m^n = m * m^{n-1}$ 
          "Exponent muss positiver Integer sein")
      )
  )
)
```

# Funktionale Sprachen: LISP

## 3. und 4. mit Iterationen

```
3.  (defun dotimes-expt (m n)
      (let ((result 1))
        (dotimes (count n result)
          (setf result (* m result))))))
```

```
4.  (defun do-expt (m n)
      (do ((result 1)
          (exponent n))
          ((zerop exponent) result)
        (setf result (* m result))
        (setf exponent (- exponent 1)))))
```

# Funktionale Sprachen: LISP

Was gibt's noch in Lisp?

1. Viele vordefinierte Funktionen:
  1. mathematische
  2. Character- und Stringfunktionen
  3. für Sequenzen, Teilsequenzen
  4. nur für Listen
  5. Input-/Output-Funktionen, formatierte Ausgabe
2. Vektoren, Arrays
3. Strukturen
4. Objekte