

# Funktionen

Auswerten eines  
funktionalen Ausdrucks  
ist das Grundkonzept

# Funktionen

- Funktionen der Mathematik sind *Vorlage*
- Beispiel:  $\sin(30^\circ) \rightarrow 0,5$   
*Funktionsname(Parameter) → Funktionswert*
- *Schachtelungen* sind möglich, werden durch Klammern gekennzeichnet und von innen nach außen ausgewertet  
 $(3 - 5)^2 \rightarrow (-2)^2 \rightarrow 4$
- Schreibweise bei Scheme: *Präfix-Notation*  
 $(\text{quadrat } (- 3 5)) \rightarrow (\text{quadrat } -2) \rightarrow 4$

# Funktionen

(	Funktionsname	Parameter	)
(	list	3 2 1 5 6 7	)
(3 2 1 5 6 7)			

- Eine neue Funktion schreiben

```
(define
  (erstes liste)
  (car liste)
)
```

# Funktionen

- Eine neue Funktion schreiben

merke dir

```
(define  
  (erstes liste)  
  (car liste)  
)
```

# Funktionen

- Eine neue Funktion schreiben

```
(define  
  (erstes liste)  
  (car liste)  
)
```

unter dem Namen

# Funktionen

- Eine neue Funktion schreiben

```
(define  
  (erstes liste)  
  (car liste)  
)
```

verwende dabei

# Funktionen

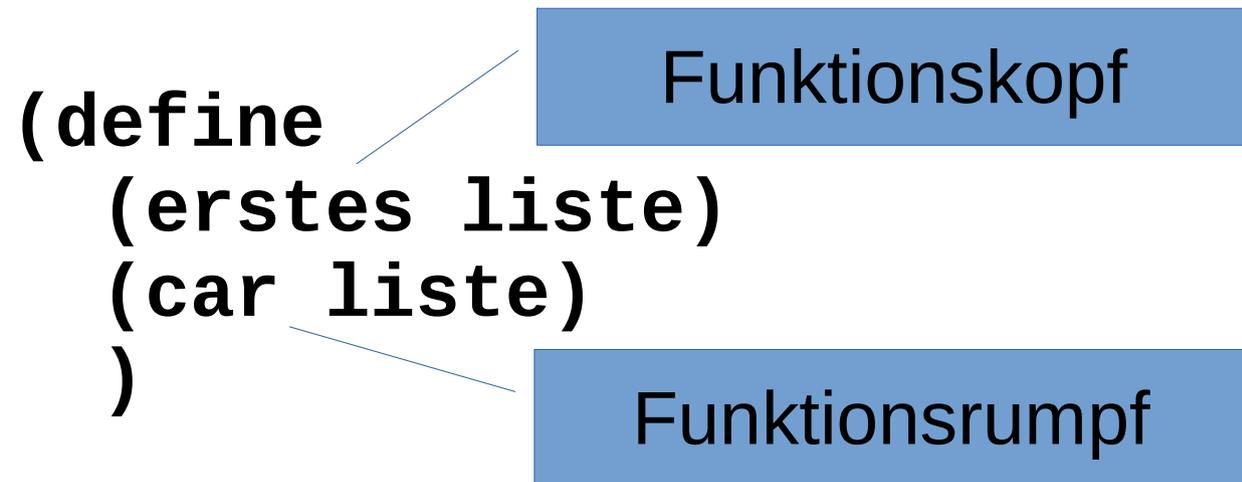
- Eine neue Funktion schreiben

```
(define
  (erstes liste)
  (car liste)
)
```

was auszuwerten ist

# Funktionen

- Eine neue Funktion schreiben



# Funktionen

- Anmerkung 1:  
Racket bietet diese und weitere Funktionen  
als Standardfunktion an:

```
(first '(1 2 3 4))  
> 1
```

```
(second '(1 2 3 4))  
> 2
```

```
(rest '(1 2 3 4))  
> (2 3 4)
```

# Funktionen

- Anmerkung 2:  
Wir gehen nicht auf die lambda-Schreibweise ein!

```
(define
  erstes
  (lambda
    (liste)
    (car liste)))
```