

Funktionen mit Scheme

Funktionen mit Scheme

Wie muss eine Funktion beschaffen sein, damit die Auswertungsumgebung versteht, was sie tun soll?

- Die Auswertungsumgebung muss wissen, **dass** sie etwas ausführen soll.

→

(<Funktionsname> ...)

Funktionen mit Scheme

Wie muss eine Funktion beschaffen sein, damit die Auswertungsumgebung etwas **lernt**?

- Die Auswertungsumgebung muss wissen, dass sie sich etwas merken soll.

→

(define ...)

Funktionen mit Scheme

Wie muss eine Funktion beschaffen sein, damit die Auswertungsumgebung weiß, wie die Funktion heißen soll?

- Der Name wird angegeben.

→

(define <name> ...)

Funktionen mit Scheme

Wie muss eine Funktion beschaffen sein, damit die Auswertungsumgebung weiß, **womit** sie arbeiten soll?

- Die Parameter werden angegeben.

→

```
(define <name> <par1> ...)
```

Funktionen mit Scheme

Wie muss eine Funktion beschaffen sein, damit die Auswertungsumgebung weiß, **was** sie tun soll?

- Der Auswertungsteil wird angegeben.

→

```
(define <name> <par1> ...  
<Auswertungsteil>)
```

Funktionen mit Scheme

- Ein **Beispiel** ...
- ... gleich in Kurzschreibweise:

```
(define
```

```
  (summe a b)
```

```
  (+ a b))
```

Funktionen mit Scheme

- ... und dann **anwenden**:

(summe 3 4)

> 7

Funktionen mit Scheme

- Falls es interessiert ...
- ... **ohne** Kurzschreibweise:

```
(define  
  summe  
  (lambda (a b)  
    (+ a b)))
```

... ist eine Funktion,
die zwei Parameter braucht,
die a und b heißen sollen.