

Eingebaute Prädikate

(Auswertung und Vergleich arithmetischer Ausdrücke)

Infixoperatoren is ::= =\= > >= < =<

```
?- X is 2-(3-4*5).
```

```
X = 19
```

```
?- sin(2.1) < exp(-0.5)+0.4.
```

```
yes
```

```
?- 2*3 =\= 1+5.
```

```
no
```

Eingebaute Prädikate (für Atome)

```
?- atom_chars(abc,X).
```

```
X = [a,b,c]
```

```
?- atom_chars(aB3c,X).
```

```
X = [a,'B','3',c]
```

```
?- atom_chars(X,[a,b,c]).
```

```
X = abc
```

Eingebaute Prädikate (für Atome)

```
?- atom_codes(abc,X).
```

```
X = [97,98,99]
```

```
?- atom_codes(aB3c,X).
```

```
X = [97,66,51,99]
```

```
?- atom_codes(X,[97,98,99]).
```

```
X = abc
```

Eingebaute Prädikate (für Atome)

```
?- atom_concat(abc,de,X).
```

```
X = abcde
```

```
?- atom_concat(X,Y,abc).
```

```
X = '', Y = abc ;
```

```
X = a, Y = bc ;
```

```
X = ab, Y = c ;
```

```
X = abc, Y = ''
```

Eingebaute Prädikate (für Atome)

```
?- atom_length(abc,X).
```

```
X = 3
```

```
?- char_code(a,X).
```

```
X = 97
```

```
?- char_code(X,97).
```

```
X = a
```

Eingebaute Prädikate (für Zahlen als atomare Terme)

```
?- number_chars(128,X).
```

```
X = ['1', '2', '8']
```

```
?- number_chars(X,['1','2','8']).
```

```
X = 128
```

```
?- number_codes(128,X).
```

```
X = [49,50,56]
```

```
?- number_codes(X,[49,50,56]).
```

```
X = 128
```

Eingebaute Prädikate (für Atome)

```
?- sub_atom(abcdef,2,3,A,S).
```

```
A = 1, S = cde
```

```
?- sub_atom(abc,Before,Length,After,Subatom).
```

```
Before = 0, Length = 0, After = 3, Subatom = '' ;
```

```
Before = 0, Length = 1, After = 2, Subatom = a ;
```

```
Before = 0, Length = 2, After = 1, Subatom = ab ;
```

```
Before = 0, Length = 3, After = 0, Subatom = abc ;
```

```
Before = 1, Length = 0, After = 2, Subatom = '' ;
```

```
Before = 1, Length = 1, After = 1, Subatom = b ;
```

```
Before = 1, Length = 2, After = 0, Subatom = bc ;
```

```
Before = 2, Length = 0, After = 1, Subatom = '' ;
```

```
Before = 2, Length = 1, After = 0, Subatom = c ;
```

```
Before = 3, Length = 0, After = 0, Subatom = ''
```

Eingebaute Prädikate (Erzeugung, Zerlegung von Termen)

```
?- arg(2,f(a,b,c,d),X).
```

```
X = b
```

```
?- functor(f(a,b,c),F,N).
```

```
F = f, N = 3.
```

```
?- functor(T,f,3).
```

```
T = f(_G278, _G279, _G280)
```


Eingebaute Prädikate (Erzeugung, Zerlegung von Termen)

?- f(a,b) =.. L.

L = [f,a,b]

?- f =.. L.

L = [f]

?- 3 =.. L.

L = [3]

?- T =.. [f,a,b].

T = f(a,b)

Eingebaute Prädikate (Unifikation zweier Terme)

```
?- f(X,b)=f(a,Y).
```

```
X = a, Y = b
```

```
?- X = Y.
```

```
X = Y
```

```
?- X = f(X).
```

Verhalten von Prolog nicht definiert

```
?- unify_with_occurs_check(X,f(X)).
```

```
no
```

Eingebaute Prädikate (Vergleich zweier Terme)

In Prolog gibt es eine Totalordnung auf der Menge der Terme.
Vergleichsprädikate sind (als Infix-Operatoren)

`== \== @< @=< @> @>=`

`?- X == X.`

`yes`

`?- X == Y.`

`no`

`?- f(a,b) @< f(b,a).`

`yes`

Eingebaute Prädikate (Liste aller Lösungen)

```
findall/3
```

```
p(a).
```

```
p(b).
```

```
?- findall(f(X,Y), (p(X),p(Y)), L).
```

```
L = [f(a,a),f(a,b),f(b,a),f(b,b)]
```

Eingebaute Prädikate (Liste aller Lösungen)

`p(a,b,d).`

`p(a,b,c).`

`p(a,b,d).`

`p(b,c,a).`

`p(b,c,f).`

`p(c,c,g).`

`?- bagof(Z, p(X,Y,Z), L).`

L=Liste aller Z mit p(X,Y,Z)

`L = [d,c,d], X = a, Y = b ;`

`L = [a,f], X = b, Y = c ;`

`L = [g], X = c, Y = c`

`?- bagof(Z, X^p(X,Y,Z), L).`

L=Liste aller Z, sodass

`L = [d,c,d], Y = b ;`

es ein X gibt mit p(X,Y,Z)

`L = [a,f,g], Y = c`

Eingebaute Prädikate (Menge aller Lösungen)

(als sortierte Liste ohne Duplikate)

`p(a,b,d).`

`p(a,b,c).`

`p(a,b,d).`

`p(b,c,a).`

`p(b,c,f).`

`p(c,c,g).`

`?- setof(Z, p(X,Y,Z), L).`

`L = [c,d], X = a, Y = b ;`

`L = [a,f], X = b, Y = c ;`

`L = [g], X = c, Y = c`

L=Menge aller Z mit p(X,Y,Z)

`?- setof(Z, X^p(X,Y,Z), L).`

`L = [c,d], Y = b ;`

`L = [a,f,g], Y = c`

L=Menge aller Z, sodass
es ein X gibt mit p(X,Y,Z)

Eingebaute Prädikate (Ein- und Ausgabe)

```
?- write(a).
```

```
a
```

```
yes
```

```
?- read(X).
```

```
f(a,b).
```

```
X = f(a,b)
```

```
?- put_char(a).
```

```
a
```

```
yes
```

```
?- get_char(X).
```

```
a
```

```
X = a
```

```
yes
```

Eingebaute Prädikate (Ein- und Ausgabe)

```
?- open(datei,write,S), write(S,a), close(S).  
S = <stream>(0x21ad050).  
?- halt.  
$ cat datei  
a
```