

# Vordefinierte Typen, Typkonstruktoren und Datenkonstruktoren

Typkonstruktor	Datenkonstruktoren	Beschreibung
Int	0,1,2,3, ...	ganze Zahlen
Integer	0,1,2,3, ...	... beliebiger Genauigkeit
Float	0.0, ...	Gleitkommazahlen
Double	0.0, ...	... doppelter Genauigkeit
Char	' ','!', ..., 'A', 'B', ...	ASCII Zeichen
Bool	True, False	Wahrheitswerte
[ - ]	[], [e <sub>1</sub> , ..., e <sub>k</sub> ]	Listen
(-, ..., -)	[], - : -	Listenkonstruktoren
- -> -	(-, ..., -)	Paare, Tupel
- -> -	(entfällt)	Funktionen

Für Konstanten des Typs [Char] kann anstelle von ['s', 't', 'r', 'i', 'n', 'g'] auch die Schreibweise "string" benutzt werden. Der Typkonstruktor -> assoziiert nach rechts, d.h. a -> b -> c ist gleichbedeutend zu a -> (b -> c).

## Vordefinierte Funktionen

Funktion	Typ	Beschreibung
not	Bool -> Bool	logische Negation
ord	Char -> Int	Codenummer im ASCII-Zeichensatz (Modul Char)
chr	Int -> Char	invers zu ord (Modul Char)
show	Show a => a -> String	Umwandlung von Druckbarem in String
error	String -> a	Erzeugt einen Laufzeitfehler
fst,snd	(a,b) -> a, (a,b) -> b	erste/zweite Komponente eines Paares
arithmetische Operationen		
abs	Num a => a -> a	Absolutbetrag
acos, asin, atan	Floating a => a -> a	Arcus Cosinus, Sinus, Tangens
atan2	RealFloat a => a -> a -> a	atan2 y x: Arcus Tangens von y/x
cos, sin	Floating a => a -> a	Cosinus, Sinus
exp, log	Floating a => a -> a	Exponentialfunktion, Logarithmus zur Basis e
negate	Num a => a -> a	Vorzeichenwechsel
pi	Floating a => a	die Zahl π
signum	Num a => a -> a	Vorzeichen
sqrt	Floating a => a -> a	Quadratwurzel

## Vordefinierte Operatoren

In der folgenden Tabelle der vordefinierten Infixoperatoren geht die Assoziativität der Operatoren aus der Spalte „A“ hervor. Ein *r* steht für rechtsassoziativ, *n* für nicht-assoziativ und *l* für linksassoziativ.

Priorität	A	Operator	Beschreibung, Typ
stark	<i>l</i>	- -	Funktionsanwendung
Infix 9	<i>r</i>	.	Funktionskomposition
	<i>l</i>	!!	Zugriff auf Listenelement
Infix 8	<i>r</i>	^	Exponentiation
Infix 7	<i>l</i>	*	Multiplikation
	<i>n</i>	/, 'div', 'quot'	Division
	<i>n</i>	'mod', 'rem'	Rest
Infix 6	<i>l</i>	+, -	Addition, Subtraktion
Infix 5	<i>r</i>	++	Listenverkettung, [a] -> [a] -> [a]
	<i>r</i>	:	Listenkonstruktor, a -> [a] -> [a]
Infix 4	<i>l</i>	'elem', 'notElem'	Elementtest, Eq a => a -> [a] -> Bool
	<i>n</i>	==, /=	Gleichheit, Ungleichheit, Eq a => a -> a -> Bool
	<i>n</i>	<, <=, >, >=	Vergleichoperatoren, Ord a => a -> a -> Bool
Infix 3	<i>r</i>	&&	logisches Und, Bool -> Bool -> Bool
Infix 2	<i>r</i>		logisches Oder, Bool -> Bool -> Bool
schwach	<i>r</i>	->	Funktionstypen
	-	=>	Kontexte
	-	(-, ..., -)	Tupelkonstruktion