

Aufbau der Funktionsdefinitionen in Haskell (von links nach rechts)

Linke Seite	Definierendes Gleichheitszeichen	Rechte Seite	Beispiel im Skript und Aufruf am Prompt (>)
Funktionsname ohne Parameter (Gross- und Kleinschreibung beachten, keine Leerzeichen, keine Sonderzeichen, keine Umlaute)	nach dem Funktionsnamen =	ein Ausdruck (der auch aus anderen Funktionsaufrufen bestehen kann)	<pre>gibeins = 1</pre> <hr/> <pre>>gibeins >1</pre>
Funktionsname mit Parameter mit keinen oder mehreren Parameternamen als Argumente, durch Leerzeichen (Applikationsoperator) getrennt	=	s.o.	<pre>verdoppel zahl = zahl * 2</pre> <hr/> <pre>>verdoppel 3 >6</pre>
oder (anstelle der Parameter): Muster (" Pattern Matching ")	=	s.o.	<pre>und 1 1 = 1 und _ _ = 0</pre> <hr/> <pre>>und 0 0 >0</pre>
ein bedingter Ausdruck mit Fallunterscheidungen (Guards). Einrückung beachten!	erst in der Guards-Zeile! =	s.o. beachte den Vergleichsoperator == im Beispiel rechts	<pre>und a b a==1 && b==1 = 1 otherwise = 0</pre> <hr/> <pre>>und 1 0 >0</pre>