

## <AusdruckL> >> <AusdruckR>

- Shiftet die Bitrepräsentation der Zahl, zu der **AusdruckL** ausgewertet (**i**), um **AusdruckR** Bits (**n**) nach **rechts**
- Verwirft die **n** **niedrigstwertigen** Bits (shiftet die niedrigstwertigen Bits nach rechts heraus)
- Höchstes bzw. Vorzeichen-Bit nicht gesetzt:**  
Fügt **n** 0-Bits als **höchstwertige** Bits vorne an (shiftet 0-Bits von links herein)
- signed-Typ und Vorzeichen-Bit gesetzt:**  
Fügt **implementationsabhängig** entweder **n** 0-Bits oder **n** 1-Bits als **höchstwertige** Bits vorne an (shiftet 0-Bits oder 1-Bits von links herein)

Variable vom Typ <b>signed char</b> (8bit)			
60 >> 1		60 >> 3	
00111100 >> 1		00111100 >> 3	
00011110 (30)		00000111 (7)	

60	>> 1	60	>> 3
00111100	>> 1	00111100	>> 3
00011110	0	00000111	100

-68	>> 1	oder	-68	>> 1
10111100	>> 1		10111100	>> 1
01011110	0 (94)		11011110	0 (-34)

	Operand AusdruckL	Operand AusdruckR	Ergebnis
<b>Konvertierung</b>	Implizite <b>unäre</b> Konvertierungen		
<b>Typen</b> (nach Konvertierung)	Beliebiger <b>integraler</b> Typ <b>TL</b>	Beliebiger <b>integraler</b> Typ <b>TR</b>	<b>TL</b> , ist kein lvalue
<b>Anmerkungen</b>	Auswertungsreihenfolge undefiniert		