

## <AusdruckL> << <AusdruckR>

- Shiftet die Bitrepräsentation der Zahl, zu der **AusdruckL** ausgewertet (**i**), um **AusdruckR** Bits (**n**) nach **links**
  - Verwirft die **n** **höchstwertigen** Bits (shiftet die höchstwertigen Bits nach links heraus)
  - Fügt **n** 0-Bits als **niedrigstwertige** Bits an (shiftet 0-Bits von rechts herein)
  - Entspricht einer **Multiplikation** von  $i * 2^n$

Variable vom Typ <code>unsigned char</code> (8bit)			
60 << 1		60 << 3	
00111100 << 1		00111100 << 3	
01111000 (120)		11100000 (224)	

60	<< 1	60	<< 3
00111100	<< 1	00111100	<< 3
001111000		001111000	

10 << 1 == 20, 10 << 3 == 80, ...
-----------------------------------

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