

Wichtige Datentypen

Datentyp	Vorzeichen	Beschreibung	auf x86
[signed] int	ja	nativer Ganzzahltyp der aktuellen Hardware, mind. so groß wie short , höchstens so groß wie long	32bit
[signed] long	ja	großer Ganzzahltyp, mindestens so groß wie int	32bit
[signed] short	ja	kleine Ganzzahltyp, mindestens so groß wie char	16bit
unsigned int	nein	wie int , jedoch ohne Vorzeichen	32bit
unsigned long	nein	wie long , jedoch ohne Vorzeichen	32bit
unsigned short	nein	wie short , jedoch ohne Vorzeichen	16bit
char	undefiniert	Character-Typ für ASCII, mindestens 8bit	8bit
signed char	ja	wie char , jedoch immer mit Vorzeichen	8bit
unsigned char	nein	wie char , jedoch immer ohne Vorzeichen	8bit
wchar_t	undefiniert	Character-Typ für Unicode, mindestens 16bit	16bit (Win32) bzw. 32bit (Unix)
bool	undefiniert	boolscher Typ, Werte sind true und false , implizite Konvertierungen von und zu Ganzzahltypen	wie char
float	ja	Gleitkommatyp	compiler-abhängig (VC: 32bit)
double	ja	Gleitkommatyp mit doppelter Genauigkeit	compiler-abhängig (VC: 64bit)

Hinweise

- Der Operator **sizeof** liefert die Größe eines Typs relativ zu **char**.
Es gilt: **sizeof(char) == 1**
- die Größen der Ganzzahltypen im Vergleich:
8bit <= char <= short <= int <= long
short < long
- Character-Typen sind Ganzzahltypen
- **signed** und **unsigned** sind Synonyme für **signed int** und **unsigned int**
- **char** ist entweder vorzeichenbehaftet oder nicht; **char**, **signed char**, und **unsigned char** sind jedoch in jedem Falle *drei* verschiedene Typen
- Benutzen Sie **typedef**!

Wichtige Operatoren

Arithmetische Operatoren

- **% + - * /**
- **%= += -= *= /=**
- **++ --**

Boolsche Operatoren

- **== != < > <= >= || && !**

Andere binäre Operatoren

- **& | ^ ~ << >>**
- **&= |= ^= ~= <<= >>=**

Sonstige Operatoren

- **=**
- **<< >> [] ()**
- *** &**
- **:: . ->**