# C++ Schlüsselwörter

## Was sind Schlüsselwörter?

Schlüsselwörter sind reservierte Bezeichner mit vordefinierter Bedeutung.
Sie können nicht als Namen für Variablen, Funktionen, Klassen usw. benutzt werden.

## Inhaltsverzeichnis



## bool

Der Datentyp bool nimmt die Werte true und false an. Bei Zuweisungen aus anderen Datentypen wird 0 zu false und jeder andere Wert zu true.

## true

Der Wert true hat so zu sagen den Wert 1. Das ist der Ergebniswert bei einer erfüllten Testbedingung. Wird einer bool-Variable ein Wert ungleich 0 zugewiesen, so erhält sie den Wert true.

## false

Der Wert false hat so zu sagen den Wert 0. Das ist der Ergebniswert bei einer nicht erfüllten Testbedingung. Wird einer bool-Variable der Wert 0 zugewiesen, so erhält sie den Wert false.

## char

char umfasst 8 Bit, also ein Byte Breite. Es hängt von den Compilereinstellungen ab ob die Zeichen intern die Werte 0-255 oder -128-127 zugeordnet werden. Welche Nummer welches Zeichen anzeigt ist system-abhängig.

## float

float nimmt Fließkommazahlen auf.

## double

double nimmt Fließkommazahlen mit erhöhter Bedeutsamkeit des Exponenten-Bereiches und der Zahlenstellen auf. long float sollte gleichbedeutend sein.

## int

int nimmt Ganz-Zahlen auf. Durch vorangestellte Modifizierer kann der Datentyp variiert werden. Dadurch sollte der Wertbereich system-abhängig sein.

## wchar_t

wchar_t wird für Zeichensätze mit mehr als 1Byte Breite benutzt.

## enum

Mit enum werden Aufzählungstypen definiert und diesen Werte aus einer Liste von ganzzahligen Konstanten zugeordnet.

## typedef

typedef weist einem Typen oder einem abgeleiteten Typen einen neuen Namen zu.

## void

void steht dort wo kein expliziter Typ angegeben werden kann. Funktionen mit dem Rückgabewert void liefern keinen Rückgabewert. Die Angabe void in der Parameterliste ist mit einer leeren Parameterliste gleichbedeutend.

## long

long ist bei den Datentypen int, float und double zulässig. Je nach System wird dadurch ein Typ mit größerem Speicherplatzbedarf und größerem Wertbereich angegeben. Nicht jede Kombination ergibt einen neuen Typ, denn z.B. ist long float gleichbedeutend mit double.

## short

short gibt bei dem Datentyp int einen kleineren Speicherplatzbedarf und einen kleineren Wertbereich an.

## signed

signed ist nur auf den Typen int und char anwendbar sowie nur mit dessen Modifikationen, also long und short, anwendbar. Auf Systemen mit binärem Datenspeicher ist somit der größte Wert nur etwa halb so groß wie beim vorzeichenlosen (unsigned) Typ, dafür gibt es aber auch negative Werte.

## unsigned

unsigned ist nur auf den Typen int und char anwendbar sowie nur mit dessen Modifikationen, also long und short, anwendbar. Auf Systemen mit binärem Datenspeicher ist somit der größte Wert ungefähr doppelt so groß wie beim vorzeichenbehafteten Typ, dafür gibt es aber auch keine negativen Werte.

## const

Mir const versehene Variablen, Zeiger oder Objekte werden als unveränderbar angesehen. Bei const Objekten können nur const Methoden angewandt werden.

## mutable

Als mutable gekennzeichnete Objekte einer Klasse können sich selbst dann ändern, wenn das Objekt als konstant markiert ist, oder eine konstante Methode abgearbeitet wird.

## volatile

Als volatile gekennzeichnete Variablen können ihre Werte auch ohne Zuweisung durch das Programm ändern, dem Compiler wird durch die Kennzeichnung mit volatile untersagt, bei der Optimierung irgendwelche Annahmen über den Variablenwert zu machen. Bei jedem lesenden Zugriff muss der Wert erneut ausgelesen werden.

## auto

auto dient zur Kennzeichnung von automatisch angelegten und beseitigten variablen im Gegensatz zu statischen. Dieser sehr veraltete Variablenmodifizierer wird kaum noch benutzt. Falls nicht anders gekennzeichnet sind alle (lokalen) Variablen automatisch.

## extern

extern zeigt an, dass die Variable oder Funktion anderswo definiert ist. Funktionen haben stets externe Bindung, sofern sie nicht als static markiert sind.

## register

register gibt dem Compiler die Empfehlung, eine häufig genutzte Variable im Prozessorregister zu halten statt im Hauptspeicher.

## static

static hat in C++ mehrere verschiedene Rollen:

1. Statische Variablen in Funktionen behalten ihren Wert auch zwischen den Funktionsaufrufen.
2. Eine statische Variable in einer Klasse ist nur einmal für die gesamte Klasse vorhanden. Eine solche Klassenvariable existiert unabhängig von der Anzahl von Objekten, die alle Zugriff auf diesen gemeinsamen Speicherplatz haben.
3. Statische Methoden können auch ohne existierende Objekte über ihren Klassennamen angesprochen werden. Statische Methoden dürfen nur statische Klassenvariablen nutzen.
4. Statische globale Funktionen und statische globale Variablen sind auf die Quelltextdatei beschränkt gültig. Solche Bezeichner sind für extern-Deklarationen aus anderen Quelltextdateien nicht sichtbar.

## Quellen

wikibooks.com
wikipedia.org
cplusplus.com
namespace-c++.de

## Hinweise

Dieser Artikel wird stetig erweitert mit Beispielen und übrigen Schlüsselwörter, bis dieser Hinweisabschnitt entfernt wird. Andere Autoren sind erwünscht.