



<https://www.artsy.net/artwork/brian-kernighan-hello-world>

Notizen

Ausgewählte Sprachelemente von C

In diesem Kapitel ...

Minimal notwendiges Rüstzeug zur Erstellung kleiner, Beispielprogramme

Dazu Vorgriff auf spätere Kapitel:

- ▶ grundlegende Programmkonstrukte
- ▶ Funktionen zur Tastatureingabe und Bildschirmausgabe.

Notizen

Variablendefinitionen

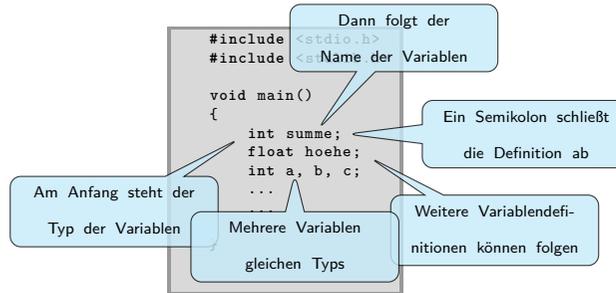


Unter einer **Variablen** verstehen wir einen mit einem Namen versehenen Speicherbereich, in dem Daten eines bestimmten Typs hinterlegt werden können.

Das im Speicherbereich der Variablen hinterlegte Datum bezeichnen wir als den **Wert** der Variablen.

Zu einer Variablen gehören also:

- ▶ ein Name
- ▶ ein Typ
- ▶ ein Speicherbereich
- ▶ ein Wert



Als Typen betrachten wir zunächst nur `int` für ganze Zahlen und `float` für Gleitkommazahlen.

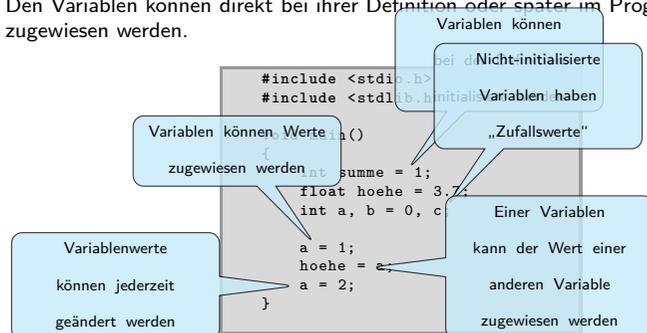
Name, Typ und Wert einer Variablen legt der Programmierer fest. Der Speicherbereich interessiert uns hier noch nicht, da er vom Compiler vergeben wird.

Notizen

Wertzuweisungen



Den Variablen können direkt bei ihrer Definition oder später im Programm **Werte** zugewiesen werden.



Die Werte können jederzeit durch erneute Zuweisung geändert werden.

Der zugewiesene Wert muss zum Typ der Variablen passen. Zum Beispiel sollten Sie einer `int`-Variablen keinen `float`-Wert zuweisen. Einer `float`-Variablen kann aber durchaus ein `int`-Wert zugewiesen werden.

Notizen

Arithmetische Operationen



Variablen und Zahlkonstanten können mit arithmetischen Operatoren verknüpft werden:

Operator	Verwendung	Bedeutung
+	$x + y$	Addition von x und y
-	$x - y$	Subtraktion von y von x
*	$x * y$	Multiplikation von x und y
/	x / y	Division von x durch y ($y \neq 0$)
%	$x \% y$	Rest bei ganzzahliger Division von x durch y (Modulo-Operator, $y \neq 0$)

Es handelt sich um die üblichen Rechenoperationen

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int summe = 1;
    float hoehe;
    int a, b, c = 0;

    hoehe = 1.2 + 2*c;
    a = b + c;
    summe = summe + 1;
}
```

Die gleiche Variable kann auf beiden Seiten einer Zuweisung vorkommen

Variablenerte können in Formeln vorkommen

Variablen vorkommen

Vorsicht bei Verwendung uninitialisierter Variablen: Ergebnis ist undefiniert

Notizen

Formelausdrücke



Mit Variablen, Zahlkonstanten, Operatoren und Klammern können Formelausdrücke gebildet werden.

```
int a;
float b, c;

a = 1;
b = (a+1)*(a+2);
c = (3.14*a - 2.7)/5;
```

Es gelten die üblichen Rechenregeln (z.B. Punktrechnung geht vor Strichrechnung).

Ganze Zahlen und Gleitkommazahlen können in Formeln durchaus gemischt vorkommen. Es wird immer so lange wie möglich im Bereich der ganzen Zahlen gerechnet. Sobald aber die erste Gleitkommazahl ins Spiel kommt, wird die weitere Berechnung im Bereich der Gleitkommazahlen durchgeführt.

Notizen

Beispiele für Fallunterscheidungen



Berechne den Absolutbetrag einer Variablen a:

```
if(a < 0)
  a = -a;
```

Wenn der Wert von a kleiner als der Wert von b ist, dann tausche die Werte von a und b:

```
if(a < b)
{
  c = a;
  a = b;
  b = c;
}
```

Weise der Variablen max den größeren der Werte von a und b zu:

```
max = a;
if(a < b)
  max = b;
```

Notizen

Vollständige Fallunterscheidung



Um eine vollständige Alternative zu programmieren arbeitet man mit if...else:

```

if( ... )
{
  ...
  ...
}
else
{
  ...
  ...
}

```

Hier steht eine **Bedingung** (zumeist ein Vergleichsausdruck)

Die hier stehenden Anweisungen werden ausgeführt, wenn die Bedingung erfüllt ist. Die hier stehenden Anweisungen werden ausgeführt, wenn die Bedingung nicht erfüllt ist.

Handelt es sich hier um eine einzelne Anweisung, können die geschweiften Klammern weggelassen werden

Dieser Teil kann vollständig fehlen. Handelt es sich hier um eine einzelne Anweisung, können die geschweiften Klammern weggelassen werden

Notizen

Beispiele für vollständige Fallunterscheidungen

Berechne das Maximum der Zahlen a und b:

```
if(a < b)
    max = a;
else
    max = b;
```

Berechne den Abstand von a und b:

```
if(a < b)
    abst = b - a;
else
    abst = a - b;
```

Notizen

Der Vergleich auf Gleichheit

Ein Vergleich auf Gleichheit wird mit dem doppelten Gleichheitszeichen durchgeführt:

```
a = 0;
...
if(a == 1)
    b = 5;
```

Das einfache Gleichheitszeichen bedeutet eine Zuweisung:

```
a = 0;
...
if(a = 1)
    b = 5;
```

Im zweiten Beispiel wird zunächst der Variablen a der Wert 1 zugewiesen. Das Ergebnis dieser Zuweisung ist 1 (wahr), sodass die nachfolgende Zuweisung (b = 5) immer ausgeführt wird.

Die Verwechslung von = und == ist einer der am häufigsten vorkommenden Fehler von Programmieranfängern.

(Dicht gefolgt vom überflüssigen Semikolon:)

```
a = 0;
...
if(a == 1);
    b = 5;
```

Notizen

Bildschirmausgabe



Um einen Text auf dem Bildschirm auszugeben, verwenden wir `printf` :

```
printf( "Dieser Text wird ausgegeben\n");
```

Der auszugebende Text wird in doppelte Hochkommata eingeschlossen. Die am Ende des Textes stehende Zeichenfolge `\n` erzeugt einen Zeilenvorschub.

Notizen

Formatierte Ausgabe



In den auszugebenden Text kann man Zahlenwerte einstreuen, indem man als Platzhalter für die fehlenden Zahlenwerte eine sogenannte Formatanweisung einfügt. Eine solche Formatanweisung besteht zum Beispiel aus einem Prozentzeichen gefolgt von dem Buchstaben `d` (für Dezimalwert). Die zugehörigen Werte werden dann als Konstanten oder Variablen durch Kommata getrennt hinter dem Ausgabertext angefügt:

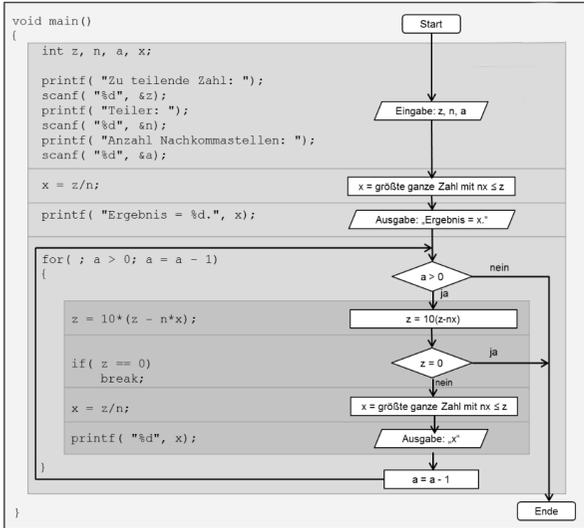
```
int wert = 1;
printf( "Die %d. Zeile hat %d Buchstaben!\n", wert, 26);
wert = 2;
printf( "Dies ist die %d. Zeile!\n", wert);
```

Ausgabe:

```
Die 1. Zeile hat 26 Buchstaben!
Dies ist die 2. Zeile!
```

Notizen

Das erste Programm



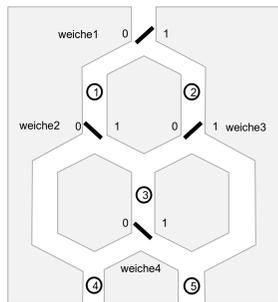
Zu teilende Zahl: 84
 Teiler: 16
 Anzahl Nachkommastellen: 4
 Ergebnis = 5.25

Zu teilende Zahl: 100
 Teiler: 7
 Anzahl Nachkommastellen: 6
 Ergebnis = 14.285714

Notizen

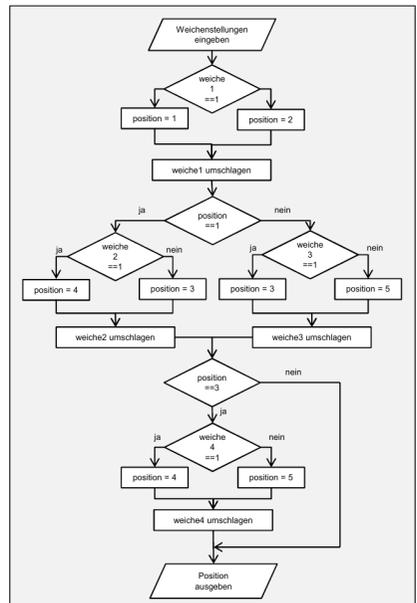
Das zweite Programm

Eine Kugel durchläuft den folgenden Parcours.



Passiert die Kugel eine Weiche, wird sie in die entsprechende Richtung gelenkt und die Weiche schlägt um. Abhängig von den Stellungen der vier Weichen (0 oder 1) erreicht die Kugel einen von zwei möglichen Ausgängen (4 oder 5).

Erstelle ein Programm, das nach Eingabe der Weichenstellungen durch den Benutzer den Ausgang berechnet.



Notizen
