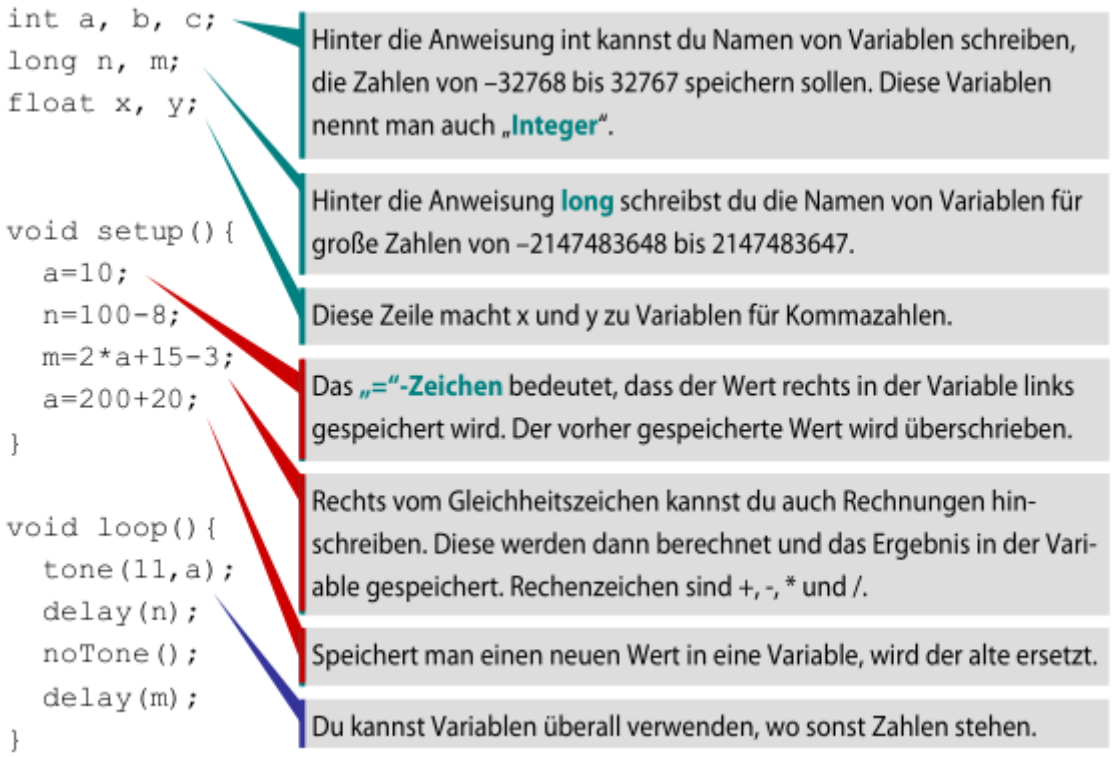
Variablen

Variablen sind **Speicherplätze**, in denen sich der Mikrocontroller Zahlenwerte merken kann. Dabei gibt es kleine Speicherplätze, die nur ganze Zahlen von -32768 bis 32767 speichern können, große Speicherplätze, die ganze Zahlen von -2147483648 bis 2147483647 speichern können und solche Speicherplätze, die Kommazahlen speichern können.

Das folgende Beispiel zeigt, wie man diese Speicherplätze *deklariert* (mit Namen versieht), wie man Zahlen darin speichert und gespeicherte wieder benutzt. Lies es aufmerksam durch, obwohl das Programm keinen echten Sinn hat:



Namensgebung: Für die Namen von Variablen kann man beliebige selbst gewählte Worte mit den gleichen Einschränkungen wie bei Unterprorammen verwenden.

Speicherplatz: Der Arduino hat in seinem Speicher insgesamt etwa 2000 Byte an Platz für Variablen. Integer benötigen jeweils 2 Byte, long oder float-Variablen jeweils 4 Byte.

Überlauf: Wenn man eine zu hohe Zahl in eine Variable speichern möchte, geschieht seltsames: In einem Integer ergibt 32767 + 1 zum Beispiel 32768. Diesen sogenannten Überlauf sollte man vermeiden.

Weitere **Fachbegriffe** im Zusammenhang mit Variablen sind:

- Deklaration: Vergabe eines Namens für eine Variable.
- Initialisierung: Erste Zuweisung eines Wertes in den deklarierten Speicherplatz.
- Manipulation: Veränderung des des Werts einer Variablen im Verlauf des Programms.

Tipp: Man kann Variablen bereits bei der Deklaration mit einem Wert versehen. Dazu schreibt man

z.B.

```
int a=8;
b=-6;
long oma=87;
p=99;
float haus=3.5;
```

Bei Kommazahlen schreibt man einen Dezimal**punkt** anstelle des Kommas.

Aufgabe 6.1

- Ordne die drei Begriffe Deklaration, Initialisierung und Manipulation passenden Zeilen im Beispielprogramm zu.
- Welchen Wert haben die Variablen a, n und m am Ende des oberen Beispielprogramms?

Aufgabe 6.2

Lege in deinem Musikstück (aus Aufgabe 5.2) Variablen für die Tondauern an, z.B. `achtel` für Achtelnoten, `viertel` für Viertel, `halbe` für Halbe und benutze sie in den Delays. So kannst du die Geschwindigkeit deines Stückes sehr leicht anpassen.

Aufgabe 6.3

Was wird das Programm unten wohl tun? Probiere es aus!

```
int f;
void setup(){
  f=100;
}

void loop(){
  tone (11,f);
  delay (100);
  f=f+1;
}
```

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
https://wiki.qg-moessingen.de/faecher:nwt:arduino:lernbaustein1:variablen:start?rev=1582116723

Last update: **19.02.2020 13:52**