

Die wichtigsten Arduino-Befehle



- Digital-Pins: 1, 2, ..., 13 (digital: die möglichen Zustände sind 0 und 1)
- Analog-Pins: A0, A1, A2, A3, A4, A5 (analog: Die möglichen Zustände sind alle Werte zwischen 0 und 1023)
- `pinMode(10, OUTPUT);` Bestimmt, dass Pin 10 als Ausgabe konfiguriert ist.
- `PinMode(A2, INPUT);` Bestimmt, dass Pin A2 als Eingabe konfiguriert ist.
- `analogRead(A0), digitalRead(13);` liest den Wert von Pin A0 bzw. 13.
- `analogWrite(A0, 512), digitalWrite(13, LOW);` schreiben auf den jeweiligen Pin
- Die gelesenen Werte dürfen verglichen oder in Berechnungen weiter benutzt werden:
 - `analogRead(A0) < 900` (ist entweder `true` (wahr) oder `false` (falsch), je nach gelesenen Wert.)
 - `digitalRead(13) == LOW` (ist der Wert am Pin 13 gleich 0? Ergebnis → `true` oder `false`)
 - `2*analogRead(A0) + 1000` (verdoppelt den gelesenen Wert und addiert 1000 dazu)
 - Die Vergleichsoperatoren sind (ähnlich wie in Rechnerblättern):
 - `<` `<=` `>` `>=`
 - `==` ist gleich
 - `!=` ist ungleich (in Rechnerblätter haben wir stattdessen `<>`)
(zum Beispiel `3 != 5` „drei ist ungleich 5“ → Wert `true` bedeutet: es ist wahr, dass 3 ungleich 5 ist)
- `//` startet einen Kommentar: alles was bis dem Zeilenvorschub folgt, wird vom Programm ignoriert.

Arduino braucht immer zwei Blöcke von Befehlen: `setup() { ... }` und `loop() { ... }`

- Im **setup**-Block wird die Anfangskonfiguration gemacht;
- Im **loop**-Block wird bestimmt, was unser Programm macht. Die darin angegebenen Befehle werden der Reihe nach unendlich oft wiederholt.