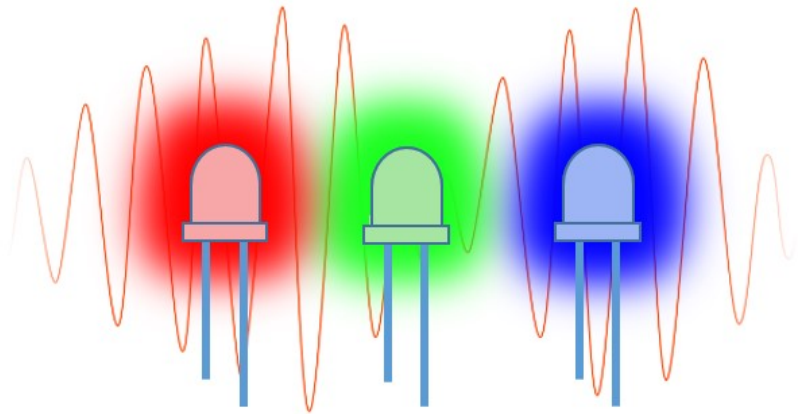


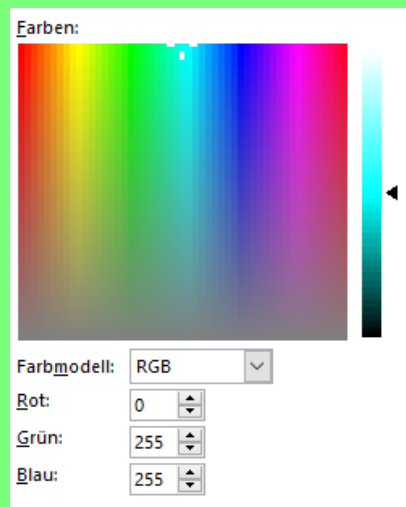
# 6



## Lösung: Erzeuge Farbmischung

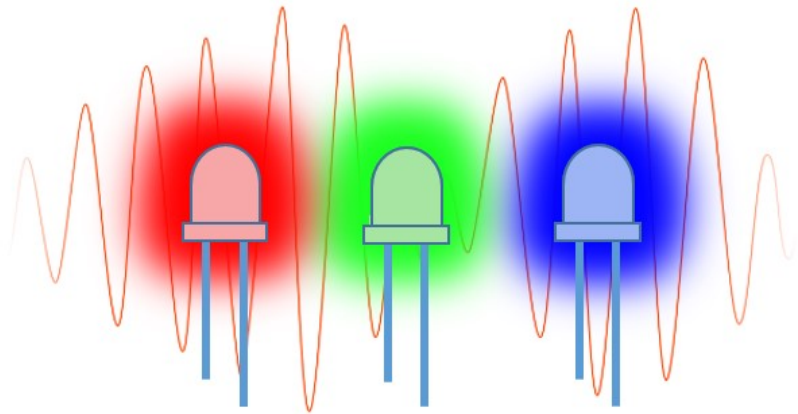
Der Color Picker wird in vielen bekannten Computerprogrammen verwendet und ist im Web auch als Onlinewerkzeug auffindbar.

Dieses RGB-Farbmischsystem nennt man Additive Farbmischung. Eine Eigenheit dieses Mischsystems ist, dass weisses Licht entsteht, wenn alle Grundfarben zusammen leuchten!



The screenshot shows a programming environment with a sidebar on the left containing categories like Agieren, Sensor, Motor, LED, Steuern, Frage, Ablauf, Dimensionieren, Variable, Mathe, Vereinfachen, Funktion, Erweitern, and Ablauf II. The main area displays three code snippets for LED control, each with a 'Hauptloop: 100'000 mal pro Sekunde' and 'Blinkcode: kurz 1 lang 1'.

- Snippet 1:** LED 1 auf 0%, LED 2 auf 100%, LED 3 auf 100%. The resulting color is cyan.
- Snippet 2:** LED 1 auf 100%, LED 2 auf 0%, LED 3 auf 100%. The resulting color is magenta.
- Snippet 3:** LED 1 auf 100%, LED 2 auf 50%, LED 3 auf 0%. The resulting color is orange.



## Lösung: Blinke mit zufälligen Farben

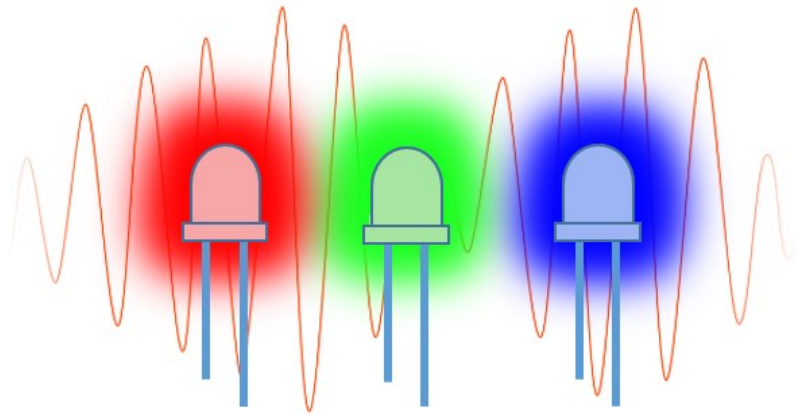
Mit diesem Programm können  $100 \times 100 \times 100 = 1$  Mio unterschiedliche Farbnuancen erzeugt werden!

Die Farbpixel auf deinem Handybildschirm bestehen ebenfalls aus 3 RGB-Lichtern und können sogar 16,7 Mio Farben darstellen! Grund ist, dass die Helligkeitsstufen normalerweise in 0 bis 255 Bit angegeben werden anstatt in Prozent.

The screenshot shows a programming environment with a dark theme. The top navigation bar includes 'Programmieren', 'Testen', 'Organisieren', and 'Hilfe'. Below this are tabs for 'Grafik', 'Code', 'Simulator', 'Programme', and '?'. On the left, a sidebar lists various components: 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', 'Erweitern', and 'Ablauf II'. The main workspace displays a program with the following blocks:

- Hauptloop:** 100'000 mal pro Sekunde (with a refresh icon), Blinkcode: kurz 1, lang 1.
- setze LED 1 auf ganzzahliger Zufallswert zwischen 0 bis 100 %**
- setze LED 2 auf ganzzahliger Zufallswert zwischen 0 bis 100 %**
- setze LED 3 auf ganzzahliger Zufallswert zwischen 0 bis 100 %**
- pausiere 1000 ms**

At the bottom right, there are icons for zooming in (+), zooming out (-), and a trash can.



## Lösung: Blende die leuchtende LED1 langsam aus

Teste das Programm im Simulator und beobachte die Helligkeit von „Rot“ an Ausgang L1.

Beachte: wenn LED1=0% ist (LED ist dunkel), wird Variable „Rot“ nicht mehr um 1 vermindert.

Dafür sorgt die Frage: „Ist Rot grösser als 0?“

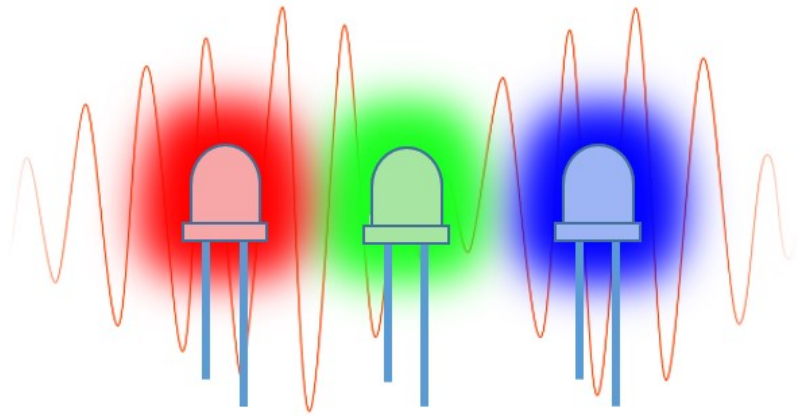
Um das Ausblenden auf der Platine neu zu starten, drücke den kleinen Reset-Knopf direkt neben dem Prozessor.

The screenshot shows the LEGO Mindstorms simulator interface. The top navigation bar includes 'Programmieren', 'Testen' (selected), 'Organisieren', and 'Hilfe'. Below this are 'Grafik', 'Code', 'Simulator' (selected), 'Programme', and '?'. The left sidebar lists various components: 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', 'Erweitern', and 'Ablauf II'. The main workspace displays a block-based program:

- Vor Hauptloop: 1x**: 'schreibe rot = 100'
- Hauptloop: 100'000 mal pro Sekunde**:
  - Condition: 'wenn rot > 0'
  - Block: 'schreibe rot = rot - 1'
  - Block: 'setze LED 1 auf rot %'
  - Block: 'pauziere 10 ms'

At the bottom, there are four LED indicators labeled M1, M2, L1, L2, L3, L4, all showing '0'.

9



## Lösung: Leuchte nach Musik und blende nach Peak aus

Teste das Programm im Simulator und klicke auf den Taster bei Sensor 4. Nach jedem Klick sollte der Pegel langsam von 100% auf 0% sinken.

Teste auch mit deiner Platine und justiere die Empfindlichkeit des Mikrofons mit einem feinen Schraubenzieher.

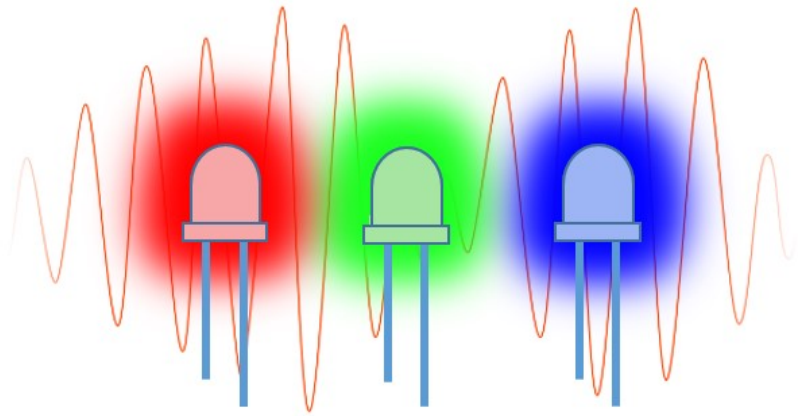
Übrigens: Wie schnell das Ausblenden geschieht, steuerst du mit der Länge der Programmpause. Anzahl Durchgänge bis Rot=0% x Pausenlänge, ergibt die Zeit für das vollständige Ausblenden.

The screenshot shows the Arduino IDE simulator interface. The top menu includes 'Programmieren', 'Testen', 'Organisieren', and 'Hilfe'. The 'Simulator' tab is active. The hardware panel shows four sensors (Sensor 1-4) with 'Schalten' and 'Regulieren' sliders, and 'Tasten / Verändern' buttons. Sensor 4's button is highlighted with a '100' value. The code block in the center is as follows:

```
Hauptloop: 100'000 mal pro Sekunde | Blinkcode: kurz 1 | lang 1
wenn [lese Wert von Sensor 4 in %] > 80
  schreibe rot = 100
  wenn [rot] > 0
    schreibe rot = rot - 1
  setze LED 1 auf [rot] %
  pausiere 10 ms
```

The status bar at the bottom shows M1 (0), M2 (0), L1 (99), L2 (0), L3 (0), and L4 (0).

10



## Lösung: Leuchte zufallsfarbig nach Musik und blende nach Peak aus

Gratuliere, du hast das erweiterte Programm für deine Lichtorgel erfolgreich geschrieben. Experimentiere jetzt mit den einzelnen Werten und passe die Effekte deinem Musikgeschmack an!

In Varianten 1-6 findest du noch weitere Ideen für interessante Lichteffekte.

The screenshot shows a programming environment with a dark theme. The top navigation bar includes 'Programmieren', 'Grafik', 'Code', 'Testen', 'Simulator', 'Organisieren', 'Programme', and 'Hilfe'. A sidebar on the left lists various categories: 'Agieren', 'Sensor', 'Motor', 'LED', 'Steuern', 'Frage', 'Ablauf', 'Dimensionieren', 'Variable', 'Mathe', 'Vereinfachen', 'Funktion', 'Erweitern', and 'Ablauf II'. The main workspace displays a code block for a 'Hauptloop: 100'000 mal pro Sekunde' with a 'Blinkcode: kurz 1 lang 1' dropdown. The code block contains the following logic:

```
when green flag clicked
  loop 100000 times
    if sensor 4 value > 80
      set red = random number 0 to 100
      set grün = random number 0 to 100
      set blau = random number 0 to 100
      wait 100 ms
      if red > 0
        set red = red - 1
      if grün > 0
        set grün = grün - 1
      if blau > 10
        set blau = blau - 1
      set LED 1 on red %
      set LED 2 on grün %
      set LED 3 on blau %
      wait 10 ms
```