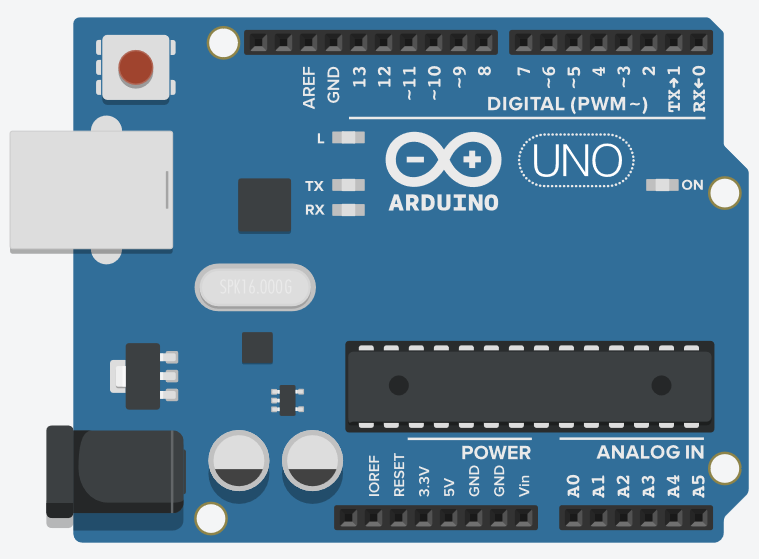
****Tinkercad Circuits Level 2:

Programmiere einen Arduino und lass eine LED Blinken

**Mit „Tinkercad Circuits“ kannst du ganz einfach virtuelle elektrische Schaltungen bauen und dabei spielerisch die Welt der Elektronik entdecken! Du brauchst dafür nichts weiter als diese Anleitung und einen Computer oder ein Tablet mit Internetzugang.**

**Für diese Anleitung solltest du bereits das Level 1 gemacht haben in dem du lernst wie du in Tinkercad Circuits Bauteile miteinander verbindest und eine Simulation startest.**

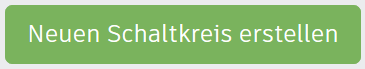
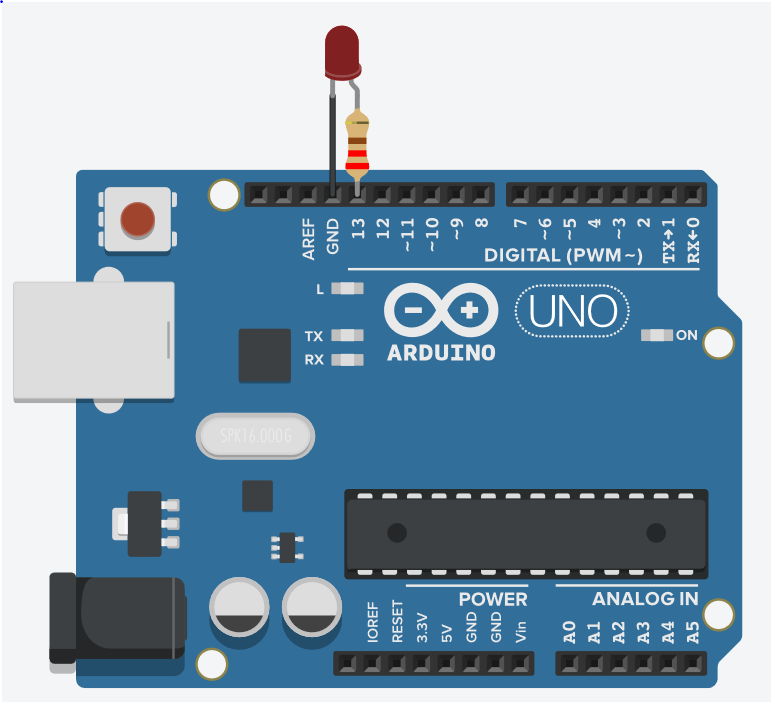
Was ist ein Arduino?



**Ein Arduino ist ein kleiner Computer den du selbst programmieren kannst.**

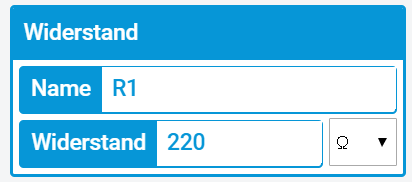
**Mit ihm kannst du z.B. LEDs oder andere Schaltungen ein-/ausschalten, Druck- und Drehknöpfe lesen, Sensordaten an deinen Computer schicken und vieles mehr – Alles was du brauchst um elektronische Musikinstrumente, nützliche Maschinen oder deine eigene Roboterarmee zu bauen!**

**So geht’s:**

1. Öffne deinen Internet-Browser und logge dich auf [www.tinkercad.com](http://www.tinkercad.com) ein.
2. Klicke links auf  und dann auf 
3. Los geht‘s! Ziehe eine **LED**, einen **Widerstand** und einen **Arduino** auf die Arbeitsfläche (siehe rechts).   
   Mit der H:\Museumspädagogik\techLAB\easy things to do\_Grafikvorlagen\Keyboard_icons\300dpi\big\r_big.png -Taste kannst du ein angeklicktes Bauteil drehen
4. Verbinde die „Anode“ (das lange Beinchen der LED) mit dem Widerstand und den Widerstand mit Pin 13 am Arduino. Die Kathode (das kurze Beinchen) wird mit einem Draht mit dem „GND“ Pin verbunden.

Siehe Level 1 falls du Hilfe beim Verbinden der Bauteile brauchst.

1. Klicke auf den Widerstand H:\Museumspädagogik\techLAB\easy things to do\P_Circuits_Level1_LED\gfx\Widerstand_freigestellt_2.png und ändere im kleinen Fenster rechts oben seinen Wert auf 220 Ω. Achte darauf, dass du die Einheit von „kΩ“ auf „Ω“



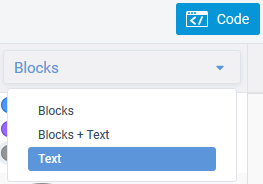
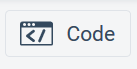
änderst.

1. Klicke auf H:\Museumspädagogik\techLAB\easy things to do\_Grafikvorlagen\Software\Tinkercad\Circuits\SimulationStarten.PNG. Die Led sollte nun Blinken.

Klicke auf H:\Museumspädagogik\techLAB\easy things to do\_Grafikvorlagen\Software\Tinkercad\Circuits\SimulationStoppen.PNG um weiter zu arbeiten.

Warum blinkt die LED?

Schauen wir in den Code um zu verstehen was passiert!

1. Klicke rechts oben auf  und wähle im drop-down Menü darunter „Text“ aus.

Jetzt siehst du rechts den Code. Er ist sozusagen die Schritt-für-Schritt Anleitung die der Arduino befolgt um die Led blinken zu lassen.

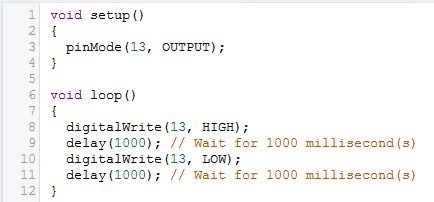
Arduino Code-Struktur

Ein Arduino-Programm besteht aus den zwei Hauptfunktionen **setup()** und **loop()**.

Die Funktionen werden durch geschwungene Klammern **{}** begrenzt.

Die Befehle innerhalb der Klammern **{}** werden nacheinander ausgeführt.

Jeder Befehl muss mit einer Strichpunkt **;** enden.

Alles was in **setup()**steht wird nach dem Einschalten nur **einmal**

ausgeführt.

Alles was in **loop()**steht wird nach **setup()**

endlos wiederholt .

Doppelte Schrägstriche // Kennzeichnen ein Kommentar.

Alles was in der Zeile danach kommt wird ignoriert.

Du kannst sie verwenden um dir im Code Notizen zu machen.

1. Klicke in das Code-Fenster um ihn zu bearbeiten.

Der Befehl:

digitalWrite(13, HIGH);

bedeutet dass der Pin 13 (an den du die LED angeschlossen hast) den Wert HIGH bekommt – der Pin wird „eingeschalten“ – die LED leuchtet.

delay(1000);

Lässt den Arduino für 1000 millisekunden (=1 Sekunde) warten.

digitalWrite(13, LOW);

setzt Pin 13 auf LOW - die LED wird ausgeschalten.

1. Ändere die Wartezeiten in delay().

z.B von delay(1000) auf delay(200) . Was Passiert? Warum?