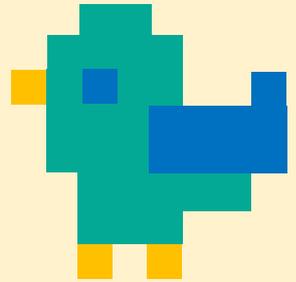


## Worum geht es?

Wie kann ein Computer farbige Bilder speichern, wenn der doch nur **0** und **1** kennt? Auf dieser Seite lernst du, wie das geht!

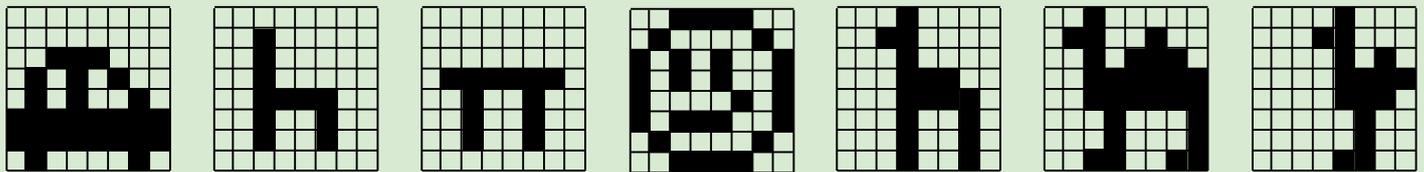


Bilder auf dem Bildschirm bestehen aus ganz vielen kleinen, quadratischen Kästchen – so genannte Pixel. Jedes Pixel hat immer genau eine Farbe. Da auf einen Bildschirm heute mehrere Millionen Pixel passen, sind sie nur noch mit der Lupe zu erkennen.

## Wie funktioniert es?

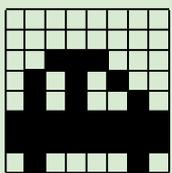
### 1. Pixelbilder zeichnen

Nimm ein Blatt Häuschenpapier und einen Bleistift. Versuche einige kleine Bilder zu zeichnen, welche nur aus leeren und komplett ausgefüllten Kästchen besteht. Hier einige Beispiele:



Ein solches Schwarz/Weiss-Bild lässt sich für den Computer ganz einfach aufschreiben. Für jedes ausgefüllte Kästchen schreiben wir eine **1** und für jedes leere Kästchen eine **0**. Schau dir das folgende Beispiel mit dem Auto an.

Kannst du ein eigenes Pixelbild mit 8 x 8 Pixeln mit **0** und **1** aufschreiben?



```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0
0 1 0 1 0 1 0 0
0 1 0 1 0 0 1 0
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
0 1 0 0 0 0 1 0
```

Zur besseren Lesbarkeit wurde die Zahlenfolge nach jeweils 8 Zahlen auf der nächsten Zeile fortgesetzt. Im Computer stehen sie aber alle direkt hintereinander. Spielt selber Computer und schickt euch gegenseitig Bilder: Schreibt ein 8 x 8 Pixel-Bild als eine lange Folge von **0** und **1** auf und tauscht diese untereinander aus. Zeichnet das zugehörige Bild und vergleicht anschliessend mit dem Original.

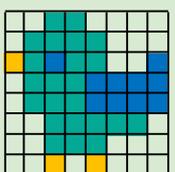
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 0
```

### 2. Von Bits und Bytes

Eine einzelne **0** oder **1** nennt man Bit. Vielleicht hast du schon einmal von 32- und 64-Bit-Computern gehört. Damit ist gemeint, wie viele Bits der Prozessor gleichzeitig verarbeiten kann. Acht Bits hintereinander ergeben ein Byte. Das Byte ist in der Welt der Computer die am häufigsten verwendete Masseinheit. Hat man sehr viele davon, spricht man von Kilobyte, Megabyte, Gigabyte oder gar Terabyte. Ein Handyfoto benötigt mehrere Millionen Byte (etwa 3-5 Megabyte pro Bild). In der Photographie spricht man häufig auch von Megapixeln, was der Anzahl Pixel pro Foto bezeichnet.

### 3. Farbige Bilder nur mit 0 und 1 aufschreiben

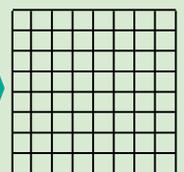
Sind Pixelbilder farbig, so reicht eine 0 und eine 1 nicht mehr aus, um für ein Pixel die jeweilige Farbe zu notieren. Wir können statt einem Bit mehrere Bits pro Pixel verwenden und in einer Farbtabelle notieren, welche Bitfolge zu welcher Farbe gehören soll. Hier ein Beispiel für ein Bild mit 4 Farben. Kannst du zur rechten Binärfolge das passende Bild in das Feld zeichnen?



```
0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 1 1 1 1 0 1 1 1 1 1 1 0 0 0 0 1 0
0 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1
0 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
```

00  
 01  
 10  
 11

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0
0 0 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0
0 0 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0
0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0
1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```



Für echte Fotos reichen 4 verschiedene Farben natürlich nicht aus. Ein einzelnes Pixel wird deshalb in den meisten Bildformaten mit 24 Bits (8 Bit pro Farbanteil **R**ot, **G**rün, **B**lau) gespeichert. Ein einziges rotes Pixel wird damit als **11111111 00000000 00000000** gespeichert – rot ist komplett vorhanden, alle anderen Farben sind leer. Durch Mischen entstehen insgesamt 16 Millionen verschiedene Farben. Ein türkises Pixel des Vogels oben hätte zum Beispiel: **00011011 10101011 10010101** (wenig Rot und viel Grün und viel Blau).