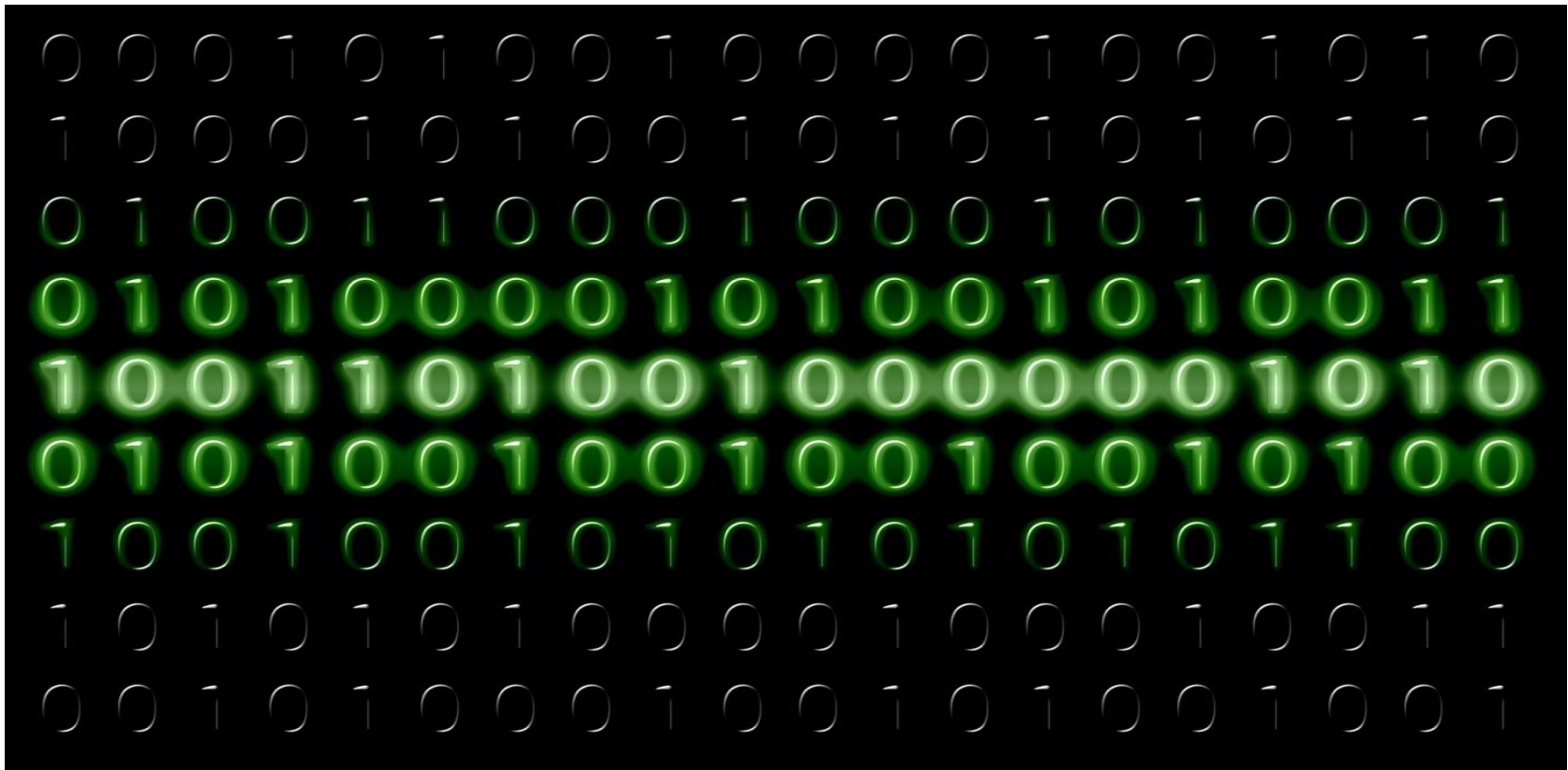


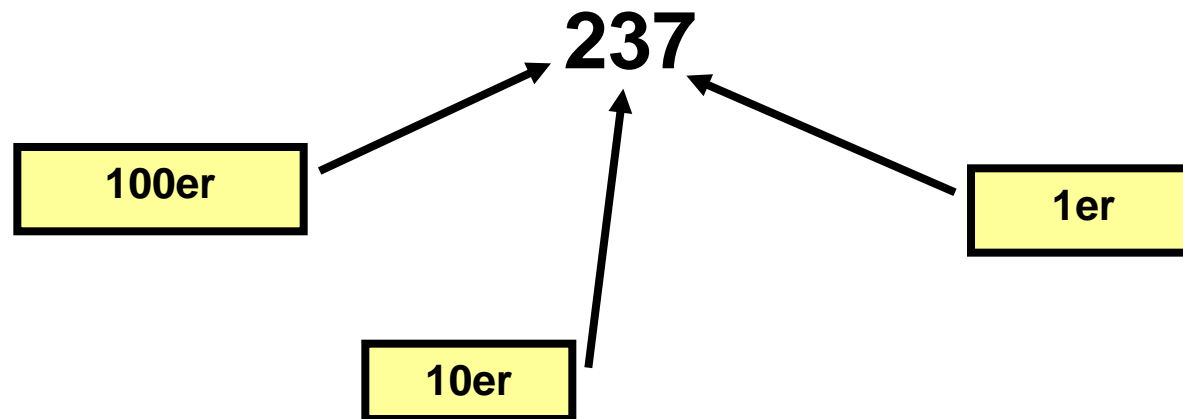
# Hexadezimale Darstellung von Zahlen



## I. Das Binärsystem

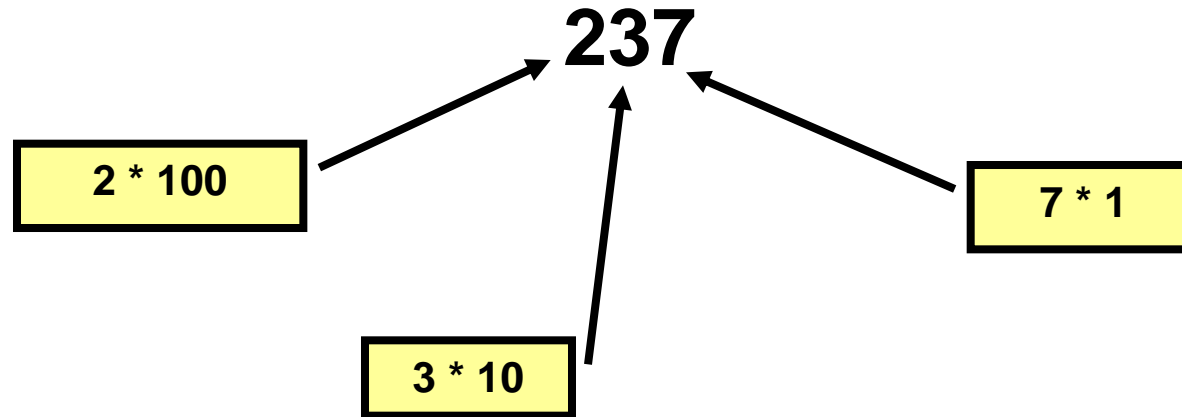
Möglicherweise haben Sie schon einmal gehört, dass Computer eigentlich nur mit den Zahlen **0 und 1** umgehen können. Während wir Menschen mit dem Dezimalsystem rechnen, verwenden Computer das so genannte Binärsystem.

Schauen wir uns erst noch mal an, wie unser Dezimalsystem funktioniert:



Was Sie hier erkennen sollen: Es ist wichtig, wo eine Ziffer steht! Ganz rechts stehen die 1er, dann kommen die 10er, dann die 100er, dann die 1.000er.....

An so einer Dezimalzahl kann man auch wunderbar ablesen, aus wie vielen 1ern, 10ern und 100ern diese Zahl besteht:

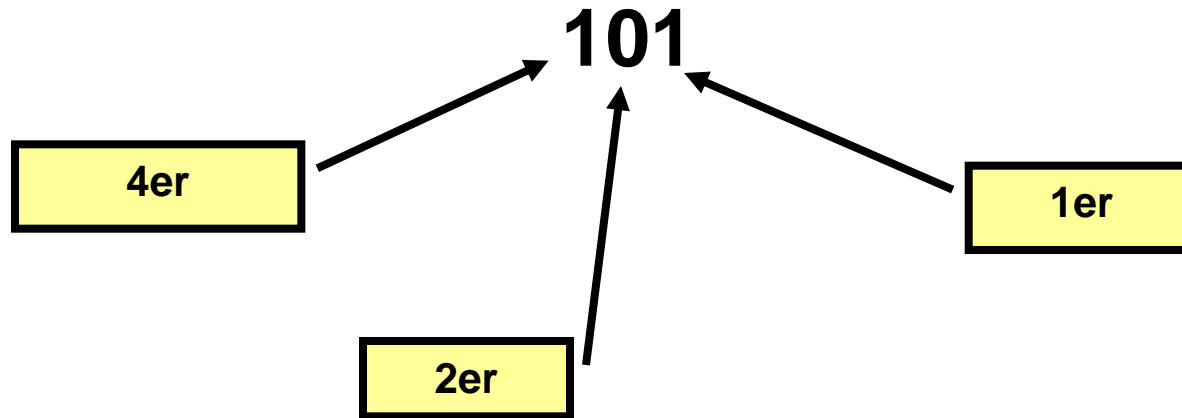


Die obige Zahl setzt sich also so zusammen:

$$237 = 7 * 1 + 3 * 10 + 2 * 100$$

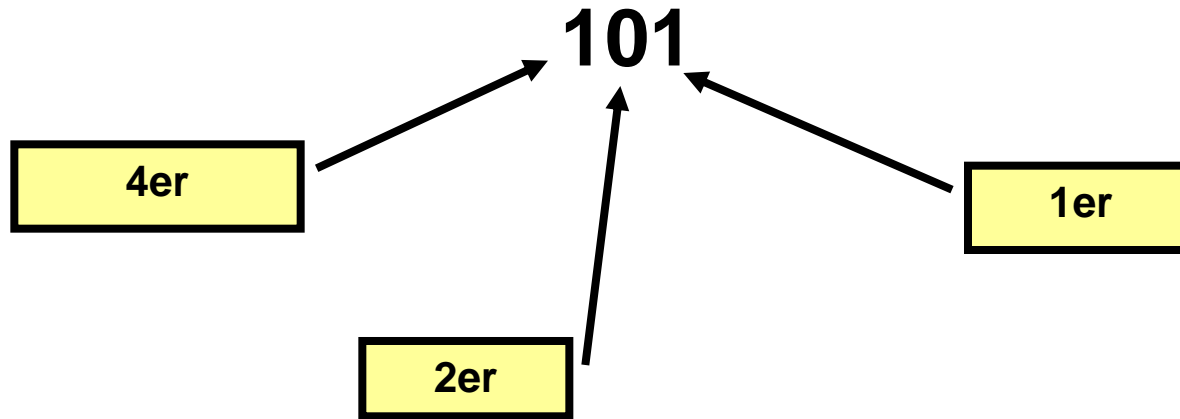
The diagram shows the equation  $237 = 7 * 1 + 3 * 10 + 2 * 100$ . Colored arrows connect the digits in the number to their respective terms in the equation: a red arrow from the '2' to  $2 * 100$ , a blue arrow from the '3' to  $3 * 10$ , and a green arrow from the '7' to  $7 * 1$ . The arrows are U-shaped, starting from the digit, going down, then across, and then up to the term.

Ganz ähnlich ist das auch im Binärsystem, das nur die Ziffern 0 und 1 kennt! Nur kommen da nach den 1ern nicht die 10er, sondern die 2er! Und danach die 4er! Dann die 8er, dann die 16er, usw...



Ist zwar ungewöhnlich, aber ansonsten genau so wie im Dezimalsystem. Jede Ziffer im Binärsystem nennt man übrigens ein **Bit**.

Schön, damit müsste es für uns möglich sein, eine Binärzahl in eine Dezimalzahl umzurechnen. Hier ein Beispiel:



$$101 = 1 * 1 + 0 * 2 + 1 * 4 = 5$$

A diagram showing the calculation of the decimal value of the binary number 101. The equation  $101 = 1 * 1 + 0 * 2 + 1 * 4 = 5$  is displayed. Colored arrows connect the digits to their respective powers of two: a red arrow from the first '1' to '1', a blue arrow from '0' to '2', and a red arrow from the second '1' to '4'. A green bracket is placed under the first '1' and its corresponding '1'.

## Aufgabe

Los geht's, rechnen Sie die folgenden Binärzahlen in Dezimalzahlen um!



$$1\ 1\ 0 = \dots$$

$$1\ 1\ 1 = \dots$$

$$1\ 0\ 1\ 0 = \dots$$

$$1\ 1\ 0\ 0\ 1\ 1\ 0\ 0 = \dots$$

$$1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 = \dots$$

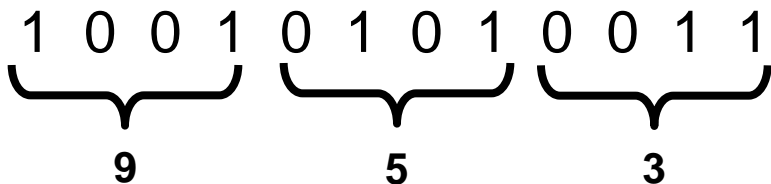
## II. Hexadezimalsystem

Früher musste man beim Computer Zahlen tatsächlich im Binärformat eingeben! Das Problem dabei: Menschen verschreiben sich hier schnell, außerdem “verstehen” Menschen Binärzahlen ganz schlecht. Das liegt daran, dass Zahlen in Binär-Darstellung einfach zu viele Ziffern haben!

### Die Lösung:

Man fasst Binärzahlen zu vierstelligen Blöcken zusammen und verwendet dafür genau eine Ziffer, die der Mensch “versteht”:

### Beispiel:



## II. Hexadezimalsystem

Hier noch mal im Überblick:

$$0000 = 0$$

$$0001 = 1$$

$$0010 = 2$$

$$0011 = 3$$

$$0100 = 4$$

$$0101 = 5$$

$$0110 = 6$$

$$0111 = 7$$

$$1000 = 8$$

$$1001 = 9$$

Genau jetzt haben wir aber ein Problem: Die nächste Binärzahl würden wir im Dezimalsystem als 10 schreiben. Das sind aber zwei Ziffern. Und das würde der Computer dann nicht für eine 10, sondern für eine 1 und eine 0 halten.



## II. Hexadezimalsystem

Die Lösung: Man erfindet zu unseren Ziffern (0, 1, 2, ....9) einfach neue Ziffern!

$$0000 = 0$$

$$0001 = 1$$

$$0010 = 2$$

$$0011 = 3$$

$$0100 = 4$$

$$0101 = 5$$

$$0110 = 6$$

$$0111 = 7$$

$$1000 = 8$$

$$1001 = 9$$

$$1010 = \mathbf{A}$$

$$1011 = \mathbf{B}$$

$$1100 = \mathbf{C}$$

$$1101 = \mathbf{D}$$

$$1110 = \mathbf{E}$$

$$1111 = \mathbf{F}$$

Damit haben Sie gerade ein neues Zahlensystem kennen gelernt. Dieses enthält folgende Ziffern:

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Da dieses Zahlensystem 16 Ziffern enthält, heißt es **Hexadezimal-System!**

*Um zu zeigen, dass es sich bei den Zahlen nicht um Dezimalzahlen handelt, kann man eine kleine 16 dazuschreiben, also z.B.  $9_{16}$ ,  $F_{16}$*

## II. Hexadezimalsystem

Ein wichtiger Begriff, den Sie garantiert schon mal gehört haben, ist das **Byte**. Ein Byte besteht aus 8 Bit. Mit unserem neuen Wissen können wir jetzt jedes Byte in Binärdarstellung als Hexadezimalzahl schreiben:

Hier ein Beispiel:

$$\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ \underbrace{\hspace{1.5em}} & & \underbrace{\hspace{1.5em}} & & & & & \\ \text{F} & & \text{A} & & & & & \end{array} = \text{FA}_{16}$$

## II. Hexadezimalsystem

Schauen wir uns jetzt noch schnell an, wie Computer, Tablets oder Smartphones Farben speichern. Dabei gehend diese Geräte so vor, wie der Fernseher. Jede Farbe wird aus den Grundfarben Rot, Grün und Blau zusammengemischt.

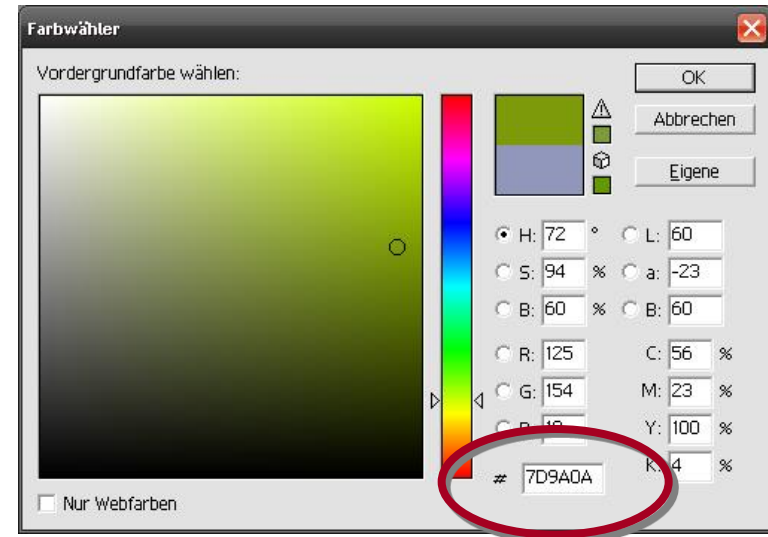
Da für jede dieser Grundfarben genau ein Byte Speicherplatz verwendet wird, sehen Farben in Hexadezimaldarstellung so aus:

**3300FF**

Rot Grün Blau

## II. Hexadezimalsystem

Damit Sie auch sehen, dass das tatsächlich so im Computer gespeichert wird, hier ein Screenshot aus dem Programm Photoshop:



Erstellen von Internet-Seiten:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Test-Seite</title>
6 </head>
7 <body text = "#000000" bgcolor="#FFFFFF">
8
9 </body>
10 </html>
```

**schwarz**      **weiß**