

# Binärcodierung

## Mit fünf Fingern anders zählen

Normalerweise kannst du mit fünf Fingern nur bis Fünf zählen. Wenn du die Zahlen jedoch anders codierst, kommst du locker bis 31. Dabei nutzt du einfach die Binärcodierung.

Kleiner Finger	Ringfinger	Mittelfinger	Zeigefinger	Daumen
$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
16er	8er	4er	2er	1er

## Beispiele

$$10001: 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 17$$

$$01111: 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 15$$

## Alle Zahlen hintereinander



**Aufgabe 1:** Sieh dir die folgende Animation an und versuche dann selbständig von 0 bis 31 mit einer Hand zu zählen.



Abbildung 1: Bild: Wolfgang Wimleitner (überarbeitet von Christian Wurzer) für [www.digikomp.at](http://www.digikomp.at)



**Aufgabe 2:**



Sobald du mit einer Hand binär zählen kannst, verwende auch die zweite Hand und setze die Zahlenfolge mit 32, 64, 128, 256 und 512 ( $2^5, 2^6, 2^7, 2^8, 2^9$ ) fort. So kannst du mit beiden Händen Zahlen bis 1023 darstellen.

## Bits und Bytes

Ein Bit ist eine Stelle einer Binärzahl und die kleinste Speichereinheit in Computern. Ein Bit kann den Wert 0 oder 1 besitzen. Ein Byte ist die zweitkleinste Speichereinheit von Computern und besteht aus 8 Bit.

	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Zahl	128er	64er	32er	16er	8er	4er	2er	1er
<b>10</b>	0	0	0	0	1	0	1	0
<b>24</b>	0	0	0	1	1	0	0	0



### Aufgabe 3:

- Schreibe die Zahlen 34, 78, 101 und 13 als Byte im Binärsystem.
- Was ist die größte Zahl, die du mit einem Byte darstellen kannst?

## Binär addieren

Man kann binär auch schriftlich addieren - dabei gilt:

$0+0 = 0$

$0+1 = 1$

$1+1 = \text{Überlauf (1 in der nächsten Stelle)}$

### Beispielrechnung:

0	0	0	0	1	0	1	0
0	0	0	1	1	0	0	0

=====  
=====

0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---



### Aufgabe 4:

- Überprüfe die Beispielrechnung, indem du die Bytes ins Dezimalsystem umrechnest.
- Addiere folgende Bytes schriftlich:  $01001011 + 00111010$  - prüfe deine Rechnung durch Umrechnung ins Dezimalsystem.

## Schwierig: Das Hexadezimalsystem (16er)

In der IT-Technik wird sehr oft das Hexadezimalsystem zur Basis 16 verwendet.

$16^3$	$16^2$	$16^1$	$16^0$
4096er	256er	16er	1er

Man braucht zum Füllen der Spalten 16 Zahlen, die man mit arabischen Ziffern allein nicht mehr darstellen kann. Daher nimmt man die Buchstaben A(10), B(11), C(12), D(13), E(14), F(15) mit dazu.

### Beispiel:

$16^3$	$16^2$	$16^1$	$16^0$
4096er	256er	16er	1er
0	A	5	E

$$0 \cdot 4096 + 10 \cdot 256 + 5 \cdot 16 + 14 \cdot 1 = \mathbf{2654}$$



### Aufgabe 5:

a) Wie musst du folgende Zahlen im Hexadezimalsystem schreiben: 10000, 675, 56, 22734?

b) Wie viele Ziffern brauchst du im Hexadezimalsystem, um den kompletten Zahlenraum eines Byte abzudecken?

Lösung === Aufgabe 5a ===  $10000_{10} = 2 \cdot 4096 + 7 \cdot 256 + 1 \cdot 16 + 0 = 2710_{16}$

$$16^3 \mid 16^2 \mid 16^1 \mid 16^0 \mid$$

4096er	256er	16er	1er
2	7	1	0

**675**<sub>10</sub>

- $0 \cdot 4096 = 0$
- $2 \cdot 256 = 512$  (Rest: 163)
- $A \cdot 16 = 160$  (Rest: 3)
- $3 \cdot 1 = 3$  (Rest: 0)

Ergebnis: **02A3**<sub>16</sub>

$16^3$	$16^2$	$16^1$	$16^0$
4096er	256er	16er	1er
0	2	A	3

**56**<sub>10</sub>

- $0 \times 4096 = 0$
- $0 \times 256 = 0$
- $3 \times 16 = 48$  (Rest: 8)
- $8 \times 1 = 8$  (Rest: 0)

Ergebnis: **0038**<sub>16</sub>

$16^3$	$16^2$	$16^1$	$16^0$
4096er	256er	16er	1er
0	0	3	8

**22734**<sub>10</sub>

- $5 \times 4096 = 20480$  (Rest: 2254)
- $8 \times 256 = 2048$  (Rest: 6)
- $0 \times 16 = 0$  (Rest: 6)
- $6 \times 1 = 6$  (Rest: 0)

Ergebnis: **5806**<sub>16</sub>

$16^3$	$16^2$	$16^1$	$16^0$
4096er	256er	16er	1er
5	8	0	6

### Aufgabe 5b

Man muss bis  $255_{10}$  kommen.

**255**<sub>10</sub>

- $F \times 16 = 240$  (Rest: 15)
- $F \times 1 = 15$  (Rest: 15)

Ergebnis: **FF**<sub>16</sub>

Man braucht für den vollen Zahlenbereich eines Byte also nur zwei Ziffern im Hexadezimalsystem.

From:  
<https://schule.riecken.de/> - Unterrichtswiki

Permanent link:  
<https://schule.riecken.de/doku.php?id=informatik:datenspuren:binaer&rev=1725540606>

Last update: **2024/09/05 12:50**

