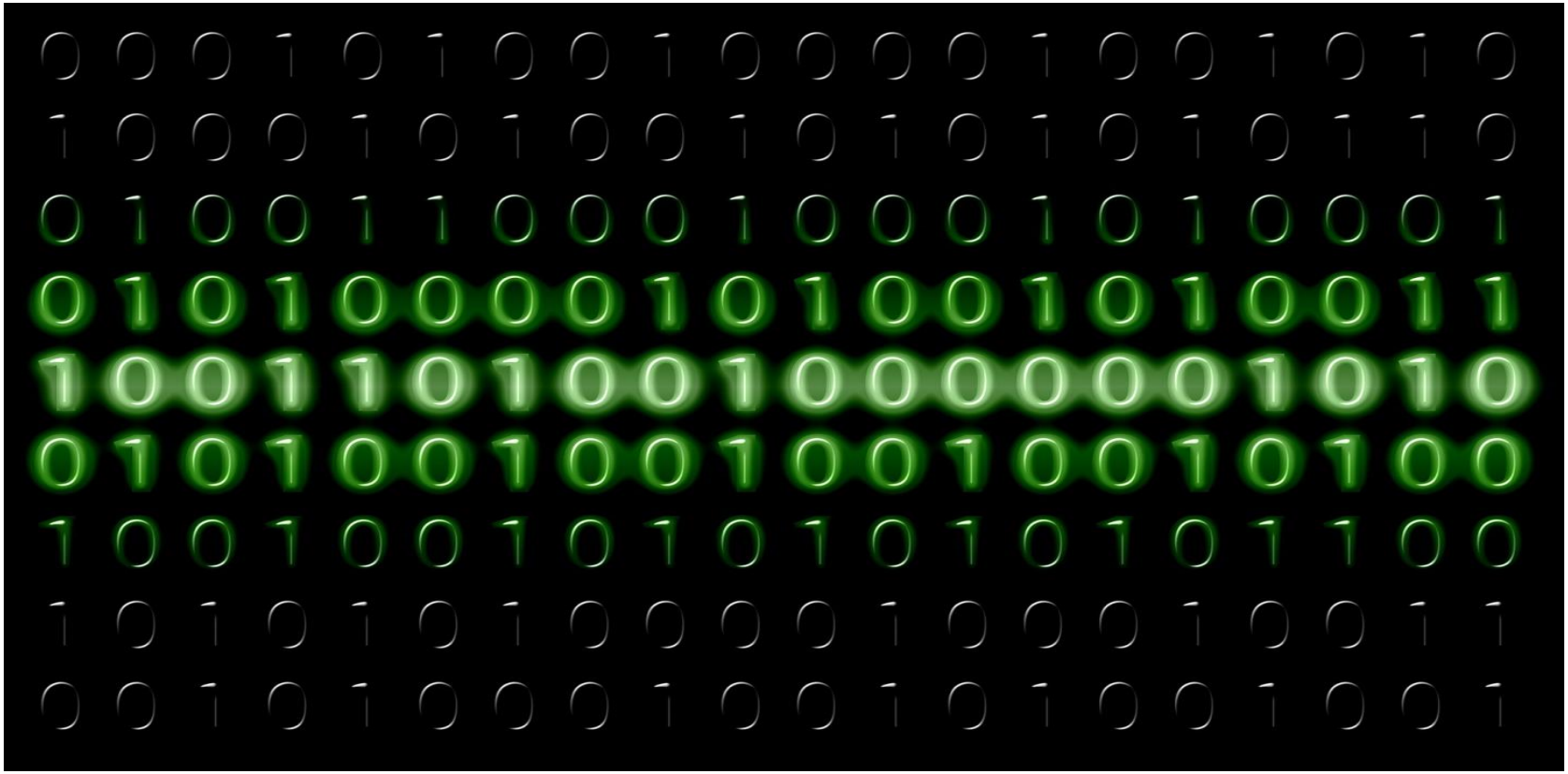
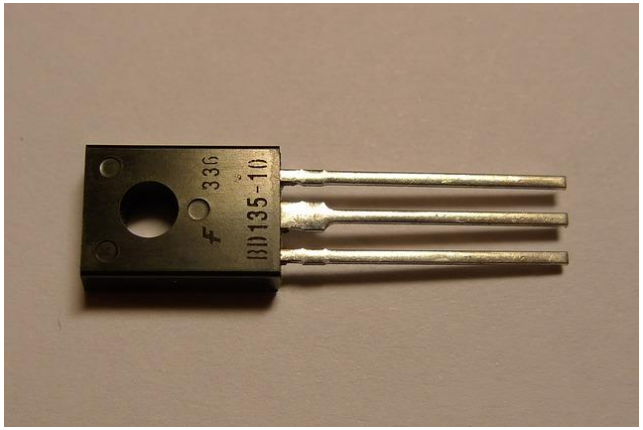
Wie können Computer eigentlich Daten speichern?



Wie funktioniert eigentlich ein Computer? Um diese Frage zu klären, muss man zuerst einmal verstehen, wie ein Computer Daten im Arbeitsspeicher (RAM) speichert. Anders ausgedrückt: Man sollte zuerst einmal verstehen, wie ein Computer sich „etwas merkt“!

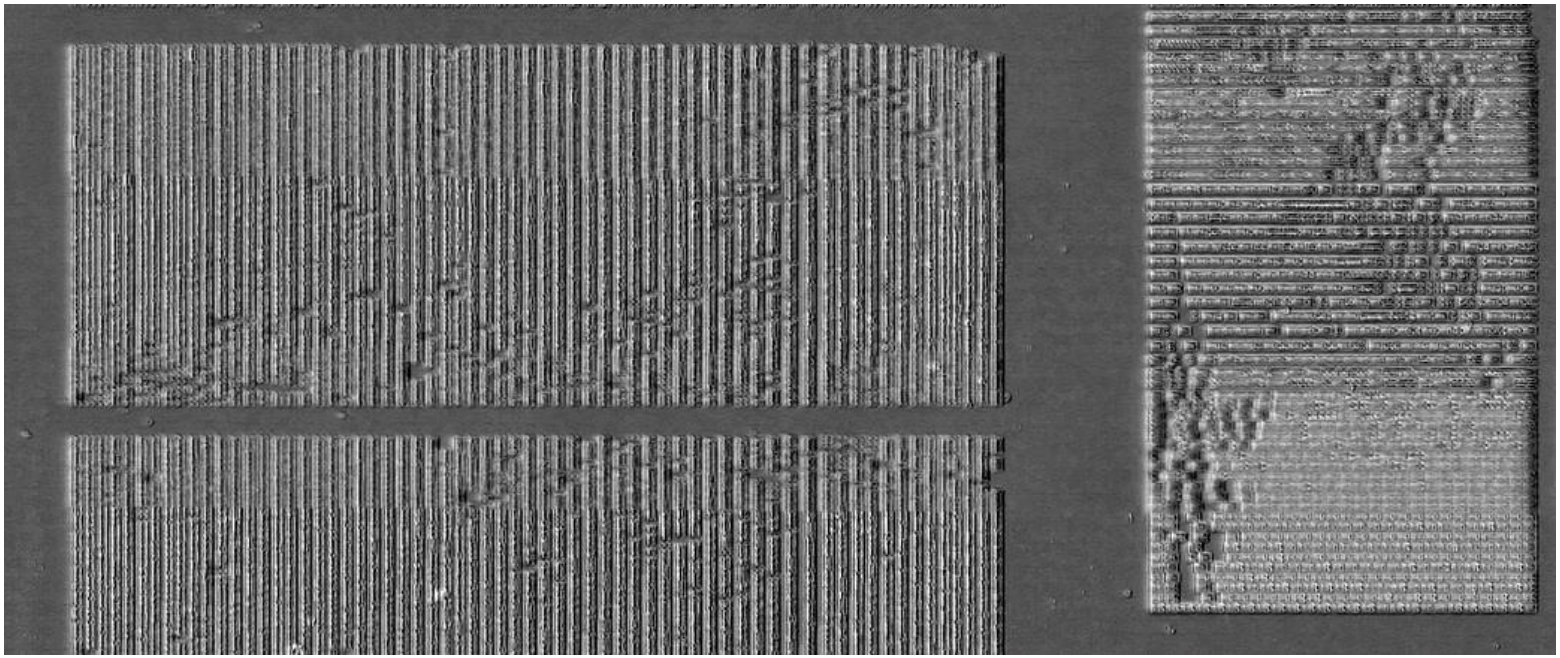
Das Wichtigste dabei: Der Arbeitsspeicher besteht aus **Transistoren**. Ein Transistor sieht so aus:



Sollten Sie schon mal einen Arbeitsspeicher gesehen haben, wundern Sie sich jetzt vielleicht. Arbeitsspeicher sehen nämlich irgendwie anders aus:

Das liegt daran, dass man inzwischen extrem kleine Transistoren bauen kann. Eigentlich werden die Transistoren nicht einmal gebaut, sondern mit Chemikalien in eine Folie geätzt. Und diese Transistoren sind dann so klein, dass auf einen Arbeitsspeicher - wie er oben zu sehen ist – locker ein paar Millionen Transistoren passen!

Auch wenn man sich das schwer vorstellen kann, dass ein paar Millionen Transistoren auf so ein kleines Bauteil passen – Unter dem Elektronenmikroskop sieht das so aus:

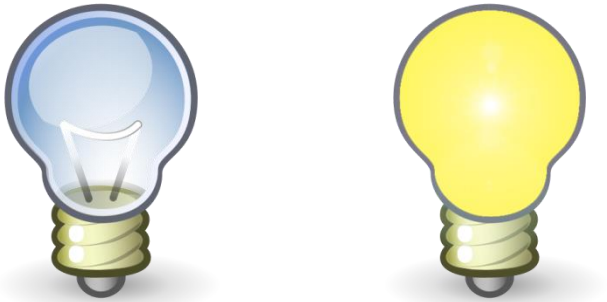


Bildquelle: By Fritzchens Fritz from Berlin [CC0], via Wikimedia Commons

Und was macht jetzt so ein Transistor?
Ganz einfach: Ein Transistor ist ein Schalter!

Das Typische an so einem Schalter: Er ist „an“ oder „aus“.

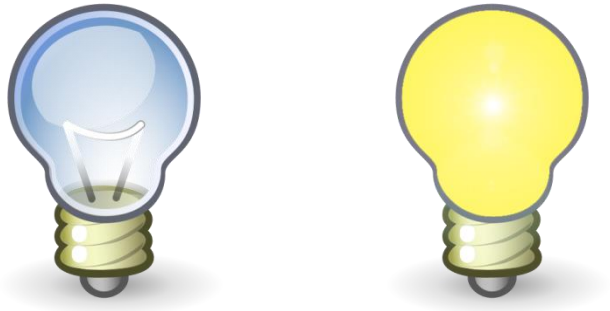
Man kann einen Transistor auch wunderbar mit einer Glühbirne vergleichen – Die ist nämlich auch an oder aus.



Aus **0**

An **1**

Wenn Sie das jetzt gar nicht so spannend finden, ist Ihnen das Wesentliche entgangen: **Damit kann man Zahlen speichern!** Das geht ganz einfach – Man legt einfach fest, dass die „brennende“ Glühbirne eine 1 sein soll und die erloschene Glühbirne eine 0. Damit können wir schon 2 Zahlen speichern. Das ist immerhin ein Anfang!



Aus
0

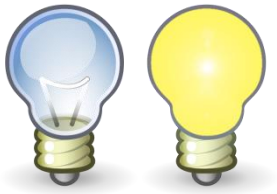
An
1

Ein guter Zeitpunkt, gleich mal den ersten wichtigen Fachbegriff zu lernen, den Sie garantiert schon gehört haben: Das **„Bit“**.

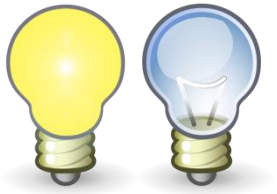
Ein Bit ist dabei die Information, die man mit einem Transistor speichern kann – also ob der Transistor an oder aus ist. Anders formuliert: Ein Bit kann die 0 für „aus“ und die 1 für „an“ speichern.

Okay, wir haben natürlich noch ein Problem. Es gibt eindeutig ein paar Zahlen mehr als nur die 0 und die 1. Aber das Problem kann man lösen: Man nimmt halt nicht nur eine Glühbirne (eigentlich einen Transistor), sondern mehrere. Schauen wir uns einfach mal an, wie viele Zahlen man mit 2 Transistoren speichern kann:

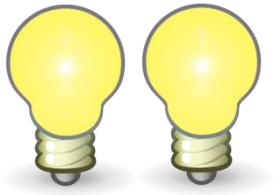
= 0



= 1



= 2

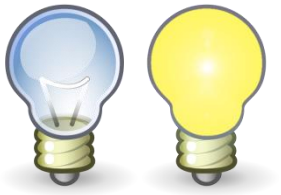


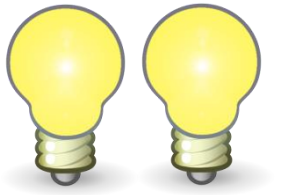
= 3

Sie sollten jetzt sehen, dass man mit 2 Transistoren schon 4 Zahlen speichern kann!

Es ist allerdings ziemlich mühsam, Glühbirnen zu malen. Das Ganze geht natürlich auch einfacher: Man kann einfach die 0 für den ausgeschalteten und die 1 für den eingeschalteten Transistor hinschreiben, das wissen Sie ja schon.

Die Darstellung mit 0 und 1 sehen Sie rechts neben den Glühbirnen!





Aufgabe 1

So, jetzt sind Sie gefordert: Wie viele Zahlen kann man denn mit 3 Transistoren darstellen?

Malen Sie sich also alle Kombinations-möglichkeiten auf ein Blatt, die es bei 3 Transistoren gibt! Fangen Sie mit den 3 ausgeschalteten Transistoren an – also: 0 0 0

Aufgabe 1

Wunderbar, hier kommt die Lösung:


= 0 0 0

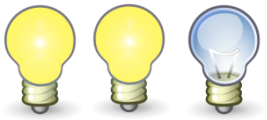

= 0 0 1


= 0 1 0


= 0 1 1


= 1 0 0


= 1 0 1


= 1 1 0


= 1 1 1

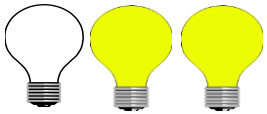
Wahrscheinlich haben Sie eine andere Reihenfolge aufgeschrieben – das ist aber kein Problem, solange Sie die selben 8 Möglichkeiten aufgeschrieben haben! Die Reihenfolge, die Sie rechts sehen, folgt einem bestimmten System, das Sie gleich kennen lernen werden.

Haben Sie gemerkt, welches System dahinter steckt? Schauen wir uns noch mal an, wie viele Zahlen wir speichern können, wenn wir eine bestimmte Anzahl an Transistoren zur Verfügung haben:

1 Transistor – 2 Zahlen
2 Transistoren – 4 Zahlen
3 Transistoren – 8 Zahlen

Jetzt müssten Sie eigentlich ein Muster erkennen und sofort sagen können, wie viel Zahlen man mit 4 Transistoren speichern kann. Na?

Die Anzahl der Zahlen verdoppelt sich pro neuem Transistor. Das heißt: Mit 4 Transistoren kann man 16 Zahlen speichern. Mit 5 Transistoren 32 Zahlen und so weiter...



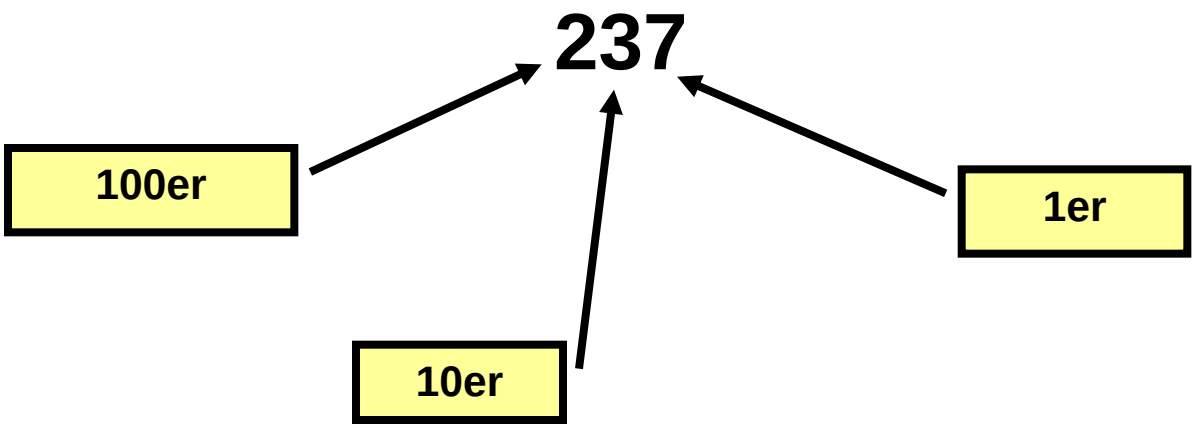
= 0 1 1

Schauen wir uns noch mal die Darstellung mit den Nullen und Einsen an, wie Sie sie oben im Bild rechts erkennen können. Diese Darstellung nennt man **Binärsystem**.

Binärsystem bedeutet – eine Art, Zahlen darzustellen, bei der man nur die Ziffern 0 und 1 verwendet. Also genau die Art, die der Computer verwendet!

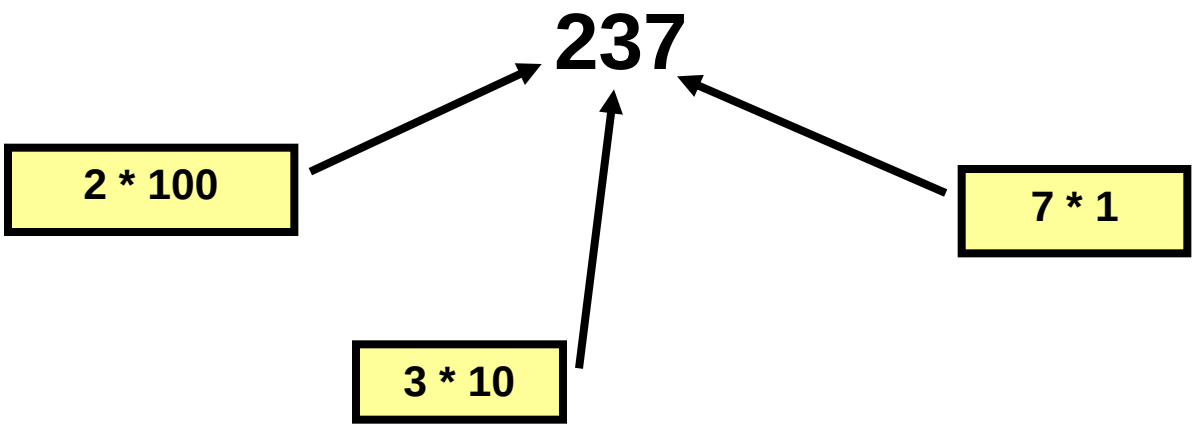
Noch mal zur Wiederholung: Der Speicher eines Computers besteht aus Transistoren. Mit einem einzigen Transistor kann man nur die Ziffern 0 oder 1 speichern. Will der Computer also eine Zahl speichern – z.B. die 6 – dann muss er das mit Nullen und Einsen tun.

Tja, Menschen denken aber nicht im Binärsystem, sondern im Dezimalsystem! Schauen wir uns mal an, wie wir eine Binärzahl in eine Dezimalzahl umrechnen. Hier kann es uns helfen, wenn wir zuerst noch mal einen Blick auf das uns bekannte Dezimalsystem werfen.

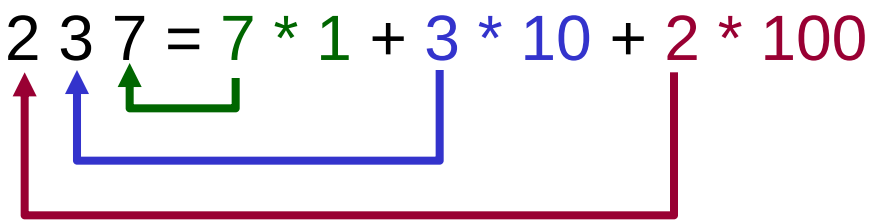


Was Sie hier erkennen sollen: Es ist wichtig, wo eine Ziffer steht! Ganz rechts stehen die 1er, dann kommen die 10er, dann die 100er, dann die 1.000er…..

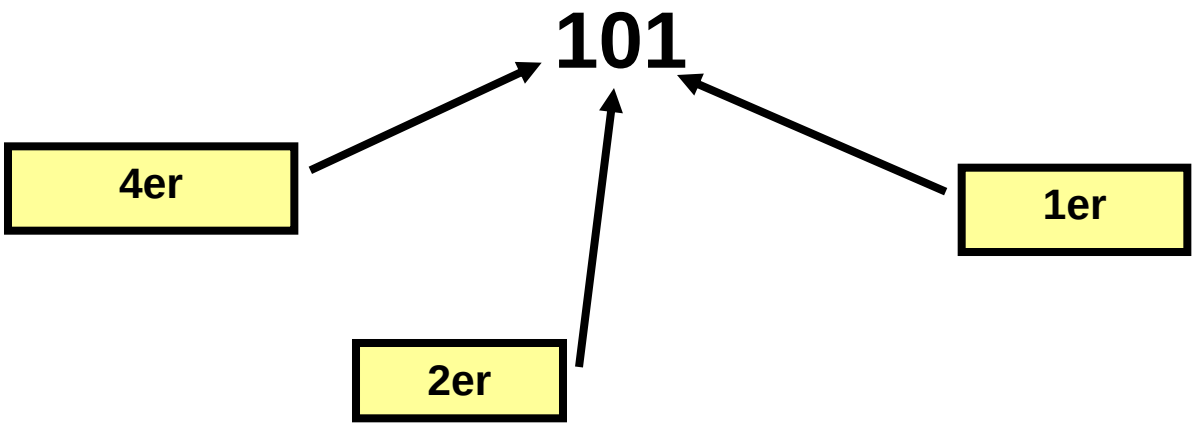
An so einer Dezimalzahl kann man auch wunderbar ablesen, aus wie vielen 1ern, 10ern und 100ern diese Zahl besteht:



Die obige Zahl setzt sich also so zusammen:

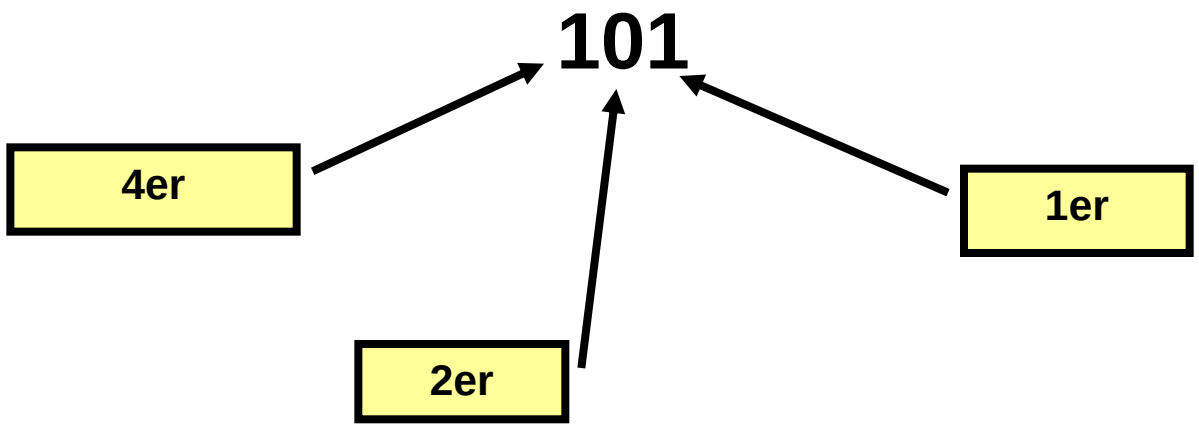


Ganz ähnlich ist das auch im Binärsystem! Nur kommen da nach den 1ern nicht die 10er, sondern die 2er! Und danach die 4er! Dann die 8er, dann die 16er, usw...



Ist zwar ungewöhnlich, aber ansonsten genau so wie im Dezimalsystem.

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



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

Aufgabe 2

Los geht‘s, schnappen Sie sich gleich mal wieder ihr Arbeitsblatt. Rechnen Sie die folgenden Binärzahlen in Dezimalzahlen um!

1 1 0 = ….

1 1 1 = ….

1 0 1 0 = ….

1 1 0 0 1 1 0 0 = ….

1 1 1 1 1 1 1 1 = ….

Vielleicht ist Ihnen aufgefallen, dass der Computer eigentlich nur Zahlen speichern kann. Warum kann ich dann mit der Tastatur auch Buchstaben und andere Zeichen eingeben?

Nun, in den Anfangstagen der Computertechnik war dies tatsächlich nicht möglich und deshalb wurden die ersten Computer auch nur zum Rechnen verwendet. Wie ein Taschenrechner eben - dieser ist ja auch nichts anderes als ein einfacher Computer.

Irgendwann kam ein kluger Mensch auf die Idee, dass man die im Computer darstellbaren Zahlen aber auch dazu nutzen könnte, Buchstaben und andere Zeichen zu verschlüsseln.

Verschlüsseln heißt: Jeder Buchstabe wird einer Zahl zugeordnet. Man könnte also z.B. festlegen, dass der Buchstabe ‘a’ im Computer als Zahl 1, der Buchstabe ‘b’ als Zahl 2 usw. gespeichert wird. Hierfür wäre es nur noch nötig, dem Computer eine Liste in die Hand zu drücken, auf der steht, welche Zahl welchem Buchstaben entspricht. Wenn der Mensch auf seiner Tastatur den Buchstaben ‘a’ drücken würde, würde der Rechner in seiner Liste nachschauen und dann die Zahl 1 in seinen Transistoren speichern.

Beispiel:

Zahl	Buchstabe
1	a
2	b
3	c

Inzwischen werden hauptsächlich **zwei** solcher Tabellen zusammen verwendet: Eine Tabelle ist der ASCII-Code (sprich: „Äskie“). Und der sieht so aus:

48	0	66	B	84	T	102	f
49	1	67	C	85	U	103	g
50	2	68	D	86	V	104	h
51	3	69	E	87	W	105	i
52	4	70	F	88	X	106	j
53	5	71	G	89	Y	107	k
54	6	72	H	90	Z	108	l
55	7	73	I	91	[	109	m
56	8	74	J	92	\	110	n
57	9	75	K	93	]	111	o
58	:	76	L	94	^	112	p
59	;	77	M	95	_	113	q
60	<	78	N	96	`	114	r
61	=	79	O	97	a	115	s
62	>	80	P	98	b	116	t
63	?	81	Q	99	c	117	u
64	@	82	R	100	d	118	v
65	A	83	S	101	e	119	w

ASCII: American Standard Code for Information Interchange

Die andere Tabelle enthält ungewöhnlichere Zeichen und heißt ANSII-Code:

Code	Zeichen	Code	Zeichen	Code	Zeichen	Code	Zeichen
0193	Á	0225	á	0211	Ó	0243	ó
0192	À	0224	à	0210	Ò	0242	ò
0194	Â	0226	â	0212	Ô	0244	ô
0195	Ã	0227	ã	0213	Õ	0245	õ
0199	Ç	0231	ç	0218	Ú	0250	ú
0201	É	0233	é	0217	Ù	0249	ù
0200	È	0232	è	0219	Û	0251	û
0202	Ê	0234	ê	0221	Ý	0253	ý
0205	Í	0237	í	0197	Å	0229	å
0204	Ì	0236	ì	0198	Æ	0230	æ
0206	Î	0238	î	0208	Đ	0240	ð
0207	Ï	0239	ï	0203	Ë	0235	ë
0209	Ñ	0241	ñ				

Die Zeichen, die in diesen beiden Listen enthalten sind, sind die Zeichen, die der Computer kennt!

Vielleicht haben Sie gemerkt, dass in den beiden Listen deutlich mehr Zeichen stehen, als Sie auf Ihrer Tastatur finden. Das liegt zum einen daran, dass in den beiden Listen auch einige so genannte „Steuerzeichen“ gespeichert werden. Das sind Befehle, die z.B. dem Drucker sagen: „Drucken!“

Andererseits gibt es aber auch viele Zeichen, die man nicht in jedem Land braucht und die deshalb nicht auf der Tastatur stehen. Das werden wir gleich mal überprüfen…

Schauen wir uns mal an, ob der Computer wirklich alle diese Zeichen kennt, die bei uns gar nicht auf der Tastatur stehen! Hierzu gibt es in Windows einen Trick, mit dem man jedes Zeichen aus einer der beiden Listen direkt aufrufen kann.

Das kann sehr praktisch sein: Stellen Sie sich z.B. vor, Sie sind in den USA und Sie wollen auf ihr Postfach (etwa bei GMX) zugreifen. Prinzipiell kein Problem – Was machen Sie aber, wenn Ihr Passwort *Döner* heißt? Das ö suchen Sie auf der amerikanischen Tastatur vergeblich…

Um auszuprobieren, wie man auf jedes beliebige Zeichen zugreifen kann, auch wenn es nicht auf der Tastatur steht, **starten** Sie am besten **Word**. Drücken Sie dazu, die **Windows-Taste** links unten, damit Sie diese Präsentation nicht beenden müssen!

- Drücken Sie die ALT-Taste und halten Sie diese gedrückt
- Geben Sie jetzt – bei gedrückter ALT-Taste den Zahlen-Code ein (auf dem Zahlenblock rechts!), den das gewünschte Zeichen hat

Beispiel:
Das kleine „a“ hat den Zahlen-Code 97. Drücken Sie also ALT und 97!

Übung

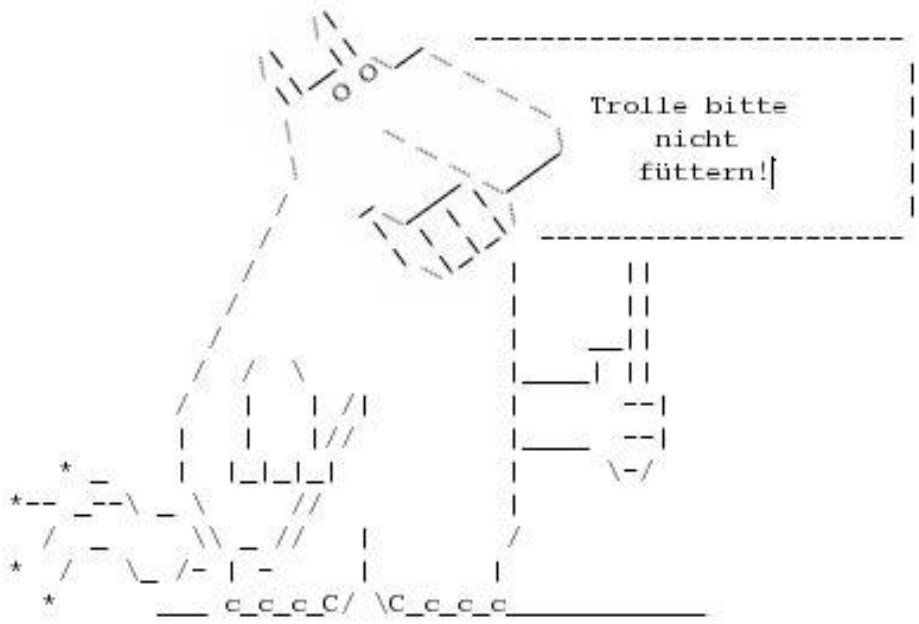
So, jetzt müssen Sie selbst rausfinden, wie man ein ö schreibt, ohne die Ö-Taste zu verwenden!

Und Probieren Sie gleich mal aus, was passiert, wenn Sie ALT und 64 drücken. Das ist nämlich auch ein sehr praktisches Zeichen…

Übrigens gibt es auch noch ein paar andere sehr interessante Dinge, die man mit diesen Zeichencodes machen kann:

Früher konnten Drucker nämlich keine Linien und Grafiken drucken. Wenn man also ein Bild auf Papier bringen wollte, konnte man nur die Zeichen verwenden, die in ASCII oder ANSII enthalten sind. Es gibt auch heute noch „Freaks“, die sich mit dieser sehr minimalistischen Art beschäftigen, Bilder auf Papier oder auf den Bildschirm zu bringen. Ein Beispiel sehen Sie auf der nächsten Folie. Am Besten wirkt das Bild, wenn Sie aufstehen und ein paar Meter vom Bildschirm weg gehen!

Wenn Sie sich dafür interessieren, dann geben Sie in eine Suchmaschine einfach mal „ASCII Art“ oder „ASCII Kunst“ ein. Inzwischen gibt es auch Programme, die aus einem echten Bild ein solches *ASCII Art* machen. Unter den Freaks gilt das aber als uncool. So was macht man mit Hand!



Betrachten wir noch einmal kurz, wie die Zeichen im Arbeitsspeicher gespeichert werden!

Als man auf die Idee kam, die Zeichen zu verschlüsseln – also jedem Zeichen eine Zahl zuzuordnen – tauchte eine wichtige Frage auf: Wie viele Transistoren bzw. wie viel Bits braucht man, um alle Buchstaben (große und kleine!), Sonderzeichen (§$%&) und ähnliches speichern zu können?

Man kam zur Überzeugung, dass 8 Transistoren eigentlich reichen müssten. Rechnen Sie schnell mal selbst nach: Wie viele Zahlen (und damit Zeichen) kann man denn mit 8 Transistoren speichern?

1 Transistor = 2 Zahlen
2 Transistoren = 4 Zahlen
3 Transistoren = 8 Zahlen
4 Transistoren = 16 Zahlen
5 Transistoren = 32 Zahlen
6 Transistoren = 64 Zahlen
7 Transistoren = 128 Zahlen
8 Transistoren = **256** Zahlen

Damit hat man also die Möglichkeit, **256 Zeichen** zu speichern. Gehen Sie am Besten gleich noch mal ein paar Folien zurück und schauen Sie nach, wie viele Zeichen tatsächlich in den beiden Listen (ASCII und ANSII) gespeichert sind!

Wenn Sie nachgeschaut haben und der Ansicht sind, es seien 255 und nicht 256 Zeichen, dann haben Sie vergessen, das Zeichen mit dem Zahlencode 0 mitzuzählen!

Es werden also 8 Transistoren verwendet, um ein Zeichen zu speichern. Diese 8 Transistoren (bzw. 8 Bit) nennt man ein **Byte**. Anders ausgedrückt: Ein Byte ist der Speicherplatz für ein Zeichen. Das sind 8 Transistoren bzw. 8 Bit.

Ein Zeichen ist natürlich ziemlich mickrig, ein einfacher Brief hat locker mehr als 1.000 Zeichen. Um tausend Zeichen speichern zu können, benötigt man folglich 1.000 Byte, das entspricht ungefähr einem Kilobyte (bzw. 1 KB).

*Sollten Sie sich gerade darüber gewundert haben, dass oben steht, 1.000 Byte seien **ungefähr** ein Kilobyte, dann sind Sie ein aufmerksamer Leser oder eine aufmerksame Leserin!*

Der Grund für dieses „ungefähr“ ist: Aus technischen Gründen werden Speicherchips üblicherweise in 2er-Potenzen hergestellt. Das heißt, dass man im Prinzip nur folgende Größen an Speicher fertigt: 2 Transistoren, 4, 8, 16, 32, 64, 128, 256, 512, 1.024, …..

Nach dieser technischen Sicht der Dinge gelten 1.024 Byte als Kilobyte. Das spielt aber bei den heutigen Speichergrößen keine Rolle mehr. Außerdem rechnen die Hardware-Hersteller inzwischen auch mit 1.000 Byte = 1 Kilobyte (KB)

Die nächstgrößeren Einheiten für die Größe des Speichers sind das **Megabyte** (bzw. MB, 1.000.000 Zeichen), das **Gigabyte** (bzw. GB, 1.000.000.000 Zeichen), das **Terrabyte**, usw.

Bei Festplatten und USB-Speichern verwendet man diese Größenangaben übrigens ebenso, obwohl dort nicht mit Transistoren gespeichert wird, sondern magnetisch (Festplatten) oder per Flash-Speicher. Wenn Sie wissen wollen, was ein **Flash-Speicher** ist: Schauen Sie am Ende dieser Stunde im Internet nach!

Warum sollte man das alles wissen?

Fast jeder benutzt das Wort **„digital“** – Aber fast niemand weiß, was das wirklich bedeutet. Obwohl wir das bisher gar nicht angesprochen haben, wissen Sie schon ungefähr, was *digital* bedeutet: Das Speichern von Daten mit 0 und 1 ist digital!

Genauer schauen wir uns das in einer der folgenden Stunden an.