

Binär-Code Alphabet (ASCII-Tabelle)

Was ist Binär-Code?

Binär-Code ist die Sprache, die Computer verwenden, um Informationen zu speichern und zu verarbeiten. Er besteht nur aus zwei Zahlen: 0 und 1. Diese nennt man Bits. Ein Computer nutzt elektrische Signale, bei denen 0 für 'kein Strom' und 1 für 'Strom fließt' steht.

Mit diesen einfachen Zuständen kann ein Computer alles darstellen: Buchstaben, Zahlen, Bilder oder Musik. Dafür verwendet man die ASCII-Tabelle, in der jedem Zeichen eine Zahl zugeordnet ist. Diese Zahl wird dann in eine Binärzahl umgewandelt. Zum Beispiel steht der Buchstabe 'A' für 65, und 65 in Binär ist 01000001.

In diesem PDF findest du die wichtigsten Zeichen (Buchstaben, Zahlen, Sonderzeichen) zusammen mit ihrem ASCII-Wert und dem passenden Binärcode.

Großbuchstaben (A-Z)

Zeichen	ASCII	Binär
A	65	01000001
B	66	01000010
C	67	01000011
D	68	01000100
E	69	01000101
F	70	01000110
G	71	01000111
H	72	01001000
I	73	01001001
J	74	01001010
K	75	01001011
L	76	01001100
M	77	01001101
N	78	01001110
O	79	01001111

Binär-Code Alphabet (ASCII-Tabelle)

P	80	01010000
Q	81	01010001
R	82	01010010
S	83	01010011
T	84	01010100
U	85	01010101
V	86	01010110
W	87	01010111
X	88	01011000
Y	89	01011001
Z	90	01011010

Kleinbuchstaben (a-z)

Zeichen	ASCII	Binär
a	97	01100001
b	98	01100010
c	99	01100011
d	100	01100100
e	101	01100101
f	102	01100110
g	103	01100111
h	104	01101000
i	105	01101001
j	106	01101010
k	107	01101011
l	108	01101100
m	109	01101101
n	110	01101110
o	111	01101111
p	112	01110000
q	113	01110001

Binär-Code Alphabet (ASCII-Tabelle)

r	114	01110010
s	115	01110011
t	116	01110100
u	117	01110101
v	118	01110110
w	119	01110111
x	120	01111000
y	121	01111001
z	122	01111010

Zahlen (0-9)

Zeichen	ASCII	Binär
0	48	00110000
1	49	00110001
2	50	00110010
3	51	00110011
4	52	00110100
5	53	00110101
6	54	00110110
7	55	00110111
8	56	00111000
9	57	00111001

Sonderzeichen

Zeichen	ASCII	Binär
!	33	00100001
"	34	00100010
#	35	00100011
\$	36	00100100
%	37	00100101
&	38	00100110

Binär-Code Alphabet (ASCII-Tabelle)

'	39	00100111
(40	00101000
)	41	00101001
*	42	00101010
+	43	00101011
,	44	00101100
-	45	00101101
.	46	00101110
/	47	00101111
:	58	00111010
;	59	00111011
<	60	00111100
=	61	00111101
>	62	00111110
?	63	00111111
@	64	01000000
[91	01011011
\	92	01011100
]	93	01011101
^	94	01011110
_	95	01011111
`	96	01100000
{	123	01111011
	124	01111100
}	125	01111101
~	126	01111110