Zeichendarstellung mit Unicode und UTF-8

ASCII – American Standard Code for Information Interchange

Dez	Hex	Sym	Dez	Hex	Sym	Dez	Hex	Sym	Dez	Hex	Sym	Dez	Hex	Sym	Dez	Hex	Sym	Dez	Hex	Sym	Dez	Hex	Sym
0	00	null	32	20	Leer-zeichen	64	40	@	96	60	`	128	80		160	A0		192	C0	À	224	E0	à
1	01		33	21	!	65	41	A	97	61	a	129	81		161	A1	¡	193	C1	Á	225	E1	á
2	02		34	22	"	66	42	B	98	62	b	130	82		162	A2	¢	194	C2	Â	226	E2	â
3	03		35	23	#	67	43	C	99	63	c	131	83		163	A3	£	195	C3	Ã	227	E3	ã
4	04		36	24	$	68	44	D	100	64	d	132	84		164	A4	¤	196	C4	Ä	228	E4	ä
5	05		37	25	%	69	45	E	101	65	e	133	85		165	A5	¥	197	C5	Å	229	E5	å
6	06		38	26	&	70	46	F	102	66	f	134	86		166	A6	¦	198	C6	Æ	230	E6	æ
7	07		39	27	'	71	47	G	103	67	g	135	87		167	A7	§	199	C7	Ç	231	E7	ç
8	08	back-space	40	28	(	72	48	H	104	68	h	136	88		168	A8	¨	200	C8	È	232	E8	è
9	09	horizontal tab	41	29	)	73	49	I	105	69	i	137	89		169	A9	©	201	C9	É	233	E9	é
10	0A	line feed	42	2A	*	74	4A	J	106	6A	j	138	8A		170	AA	ª	202	CA	Ê	234	EA	ê
11	0B		43	2B	+	75	4B	K	107	6B	k	139	8B		171	AB	«	203	CB	Ë	235	EB	ë
12	0C		44	2C	,	76	4C	L	108	6C	l	140	8C		172	AC	¬	204	CC	Ì	236	EC	ì
13	0D	carriage return	45	2D	-	77	4D	M	109	6D	m	141	8D		173	AD	-	205	CD	Í	237	ED	í
14	0E		46	2E	.	78	4E	N	110	6E	n	142	8E		174	AE	®	206	CE	Î	238	EE	î
15	0F		47	2F	/	79	4F	O	111	6F	o	143	8F		175	AF	¯	207	CF	Ï	239	EF	ï
16	10		48	30	0	80	50	P	112	70	p	144	90		176	B0	°	208	D0	Ð	240	F0	ð
17	11		49	31	1	81	51	Q	113	71	q	145	91		177	B1	±	209	D1	Ñ	241	F1	ñ
18	12		50	32	2	82	52	R	114	72	r	146	92		178	B2	²	210	D2	Ò	242	F2	ò
19	13		51	33	3	83	53	S	115	73	s	147	93		179	B3	³	211	D3	Ó	243	F3	ó
20	14		52	34	4	84	54	T	116	74	t	148	94		180	B4	´	212	D4	Ô	244	F4	ô
21	15		53	35	5	85	55	U	117	75	u	149	95		181	B5	µ	213	D5	Õ	245	F5	õ
22	16		54	36	6	86	56	V	118	76	v	150	96		182	B6	¶	214	D6	Ö	246	F6	ö
23	17		55	37	7	87	57	W	119	77	w	151	97		183	B7	·	215	D7	×	247	F7	÷
24	18		56	38	8	88	58	X	120	78	x	152	98		184	B8	¸	216	D8	Ø	248	F8	ø
25	19		57	39	9	89	59	Y	121	79	y	153	99		185	B9	¹	217	D9	Ù	249	F9	ù
26	1A		58	3A	:	90	5A	Z	122	7A	z	154	9A		186	BA	º	218	DA	Ú	250	FA	ú
27	1B		59	3B	;	91	5B	[	123	7B	{	155	9B		187	BB	»	219	DB	Û	251	FB	û
28	1C		60	3C	<	92	5C	\	124	7C	|	156	9C		188	BC	¼	220	DC	Ü	252	FC	ü
29	1D		61	3D	=	93	5D	]	125	7D	}	157	9D		189	BD	½	221	DD	Ý	253	FD	ý
30	1E		62	3E	>	94	5E	^	126	7E	~	158	9E		190	BE	¾	222	DE	Þ	254	FE	þ
31	1F		63	3F	?	95	5F		127	7F		159	9F		191	BF	¿	223	DF	ß	255	FF	ÿ

In der Mittelstufe wurde die Codierung von Text mithilfe des ASCII-Standards besprochen. Hierbei wird jedem Zeichen ein Wert zwischen 0 und 255 (8 Bit) zugewiesen. Oben siehst du die ASCII-Codetabelle, leere Zellen enthalten Steuerzeichen, welche für die Darstellung am PC nötig waren. Die wichtigsten Steuerzeichen sind in der Tabelle beschrieben.

In einem früheren, hauptsächlich in Amerika benutzten Standard waren lediglich die Zeichen von 0 bis 127 definiert, das letzte, achte Bit wurde zur Fehlerüberprüfung verwendet. Erst später wurde das 8. Bit dazu genommen, um weitere Zeichen, wie z.B. die deutschen Umlaute codieren zu können.

(A1)

Wandle die nachfolgenden Wörter, die in Hexadezimal-Darstellung vorliegen, in lesbaren Text um:

1. `49 6E 66 6F 72 6D 61 74 69 6B`
2. `42 69 6E E4 72`
3. `43 6F 6D 70 75 74 65 72`

Mit einer 8-Bit-Codierung lassen sich nicht mehr Zeichen darstellen, was insbesondere bei anderen Sprachen – wie z.B. griechisch – andere Codierungen nötig machte. Da in diesen Sprachen jedoch die bei uns gebräuchlichen Umlaute nicht benötigt werden, wurde der durch das 8. Bit hinzugekommene Block vom Zeichen 128 bis 255 für die dortigen Zeichen verwendet. Diese und andere länderspezifischen Codierungen lassen sich z.B. unter https://de.wikipedia.org/wiki/ISO_8859 nachschauen.

(A2)

Welche der obigen Wörter würden mit den griechischen Zeichensatz falsch dargestellt werden und warum?

Unicode

Um Probleme, die sich zum einen mit unterschiedlichen Zeichensätzen, zum anderen auch durch andere Sprachen, die mehr als 128 Zeichen umfassen, ergeben haben, wurde der Unicode-Standard entwickelt.

> Unicode ist ein internationaler Standard, in dem langfristig für jedes sinnvolle Schriftzeichen oder Textelement aller bekannten Schriftkulturen und Zeichensysteme ein digitaler Code festgelegt wird. Ziel ist es, die Verwendung unterschiedlicher und inkompatibler Kodierungen in verschiedenen Ländern oder Kulturkreisen zu beseitigen. Unicode wird ständig um Zeichen weiterer Schriftsysteme durch das Unicode-Konsortium ergänzt. (Wikipedia, https://de.wikipedia.org/wiki/Unicode)

Im Unicode Standard hat jedes Zeichen einen eigenen "Unicode-Code", damit lassen sich derzeit 1.111.998 elementare Zeichen ("Codepunkte") abbilden. Darstellung: U+00DF (Mindestens 4x4Bit, bis zu U+10FFFF) Diese Codepunkte bilden den Unicode Zeichensatz.

Die Zeichen des Zeichensatzes werden wiederum auf unterschiedliche Weisen codiert, beispielsweise in Betriebssystemen. Wir betrachten die UTF-8 Kodierung von Unicode Zeichen genauer.

UTF-8 Implementation des Unicode Zeichensatzes

Hier kann ein einzelnes Zeichen in der UTF-8-Codierung bis zu 4 Bytes umfassen, nach folgenden Regeln:

- Ist die Binärdarstellung des Unicode-Codes nicht länger als ein Byte und das das erste Bit eine

0, werden die restlichen 7 Bit gemäß des ASCII Codes verwendet, die 128 verbleibenden Möglichkeiten entsprechen also genau dem ASCII-Code.
- Ist die Binärdarstellung des Unicode-Codes länger als ein Byte oder der Code ist ein Byte lang und beginnt mit einer 1 geht man wie folgt vor: Der Unicode-Code wird in 6 Bit lange Teile aufgeteilt. Für jedes dieser 6 Bit Pakete wird ein Byte zur Darstellung verwendet, jedes Byte beginnt mit '10'. Das erste Byte beginnt mit einer '1' für jedes Byte, das verwendet wird. Benötigt man also 3 Byte, um ein Zeichen in UTF-8 darzustellen, beginnt das erste Byte mit '111'. Bevor die Nutzdaten beginnen, muss noch eine Null eingefügt werden [1)]

Beispiele:

(1)

$y = 79_{16} = 0111\ 100_2$

Beginnt mit einer Null und ist nicht länger als ein Byte → die letzten 7Bit werden verwendet, um zu codieren, also ein "ASCII k" in UTF-8

UTF-8: 0110 1011

(2)

$ä = E4_{16} = 1110\ 0100_2$

Nur ein Byte lang, beginnt aber mit einer 1. Die 8 Bit müssen in 6 Bit Abschnitte geteilt und auf 2 Byte verteilt werden, beginnend von rechts, links wird stets mit 0en aufgefüllt:

`000011 100100`

- Das zweite Byte beginnt nach den Regeln mit 10, daran schließen die Nutzdaten an: `10 100100`
- Das erste Byte beginnt mit 11 (weil man zwei Byte benötigt) dann wird mit 0en aufgefüllt, dann kommen die Nutzedaten: `11 000011`

Die UTF-8 Codierung des Unicode-ä ist also `1100 0011 1010 0100`. Die Nutzdaten, die den Code des Unicode Zeichens transportieren sind in jeden Byte nur die jeweils letzten 6 Bit.

(3) 乐 → U+4E50 → $4E50_{16}$ → 0100 1110 0101 0000_2

Chinese, Japanese, Korean (cjk) unified ideograph (U+4E50)

- 16 Bit Daten zu codieren, dafür braucht man 3 Byte (3 x 6 = 18)
- Der UTF-8 Code beginnt also mit der Startsequenz 111
- Dann von rechst beginnend 6 Bit (01 000), das Byte beginnt mit 10 (Regel) also ist das dritte Byte `1010 1000`
- Die nächsten 6 Bit analog: `1110 01` → `1011 1001`

- Die fehlenden 4 Bit `0100` mit Padding + Startsequenz (111)ergeben das erste Byte `1110 0100`

Die UTF-8 Codierung des Unicode-Zeichens 乔 ist also 3 Byte lang und sieht so aus: `1110 0100 1011 1001 1010 0000`

(A3)

Wandle die nachfolgenden Zeichen des Unicode Zeichensatzen in die UTF-8-Codierung um. Der Hexadezimalcode des Unicode Zeichens ist jeweils angegeben.

Gehe jeweils wie in den Beispielen oben vor. Markiere die "Nutzdaten" die das eigentlich Unicode-Zeichen "transportieren".

1. I=49_{16}
2. Ö=$D6_{16}$
3. 弈=$5F08_{16}$
4. 𠁊=$1F60A_{16}$

Lösung 1

0**1001001**, ein Byte, erstes Bit 0.

Lösung 2

110**00011** 10**010110**

Lösung 3

1110**0101** 10**111100** 10**001000**

Lösung 4

1111**0000** 10**011111** 10**011000** 10**001010**

(A4)

Wie viele unterschiedliche Unicode-Zeichen lassen sich theoretisch mit 1 Byte, 2 Bytes, 3 Bytes und 4 Bytes unter Beachtung der UTF-8-Regeln darstellen?

Lösung

- 1 Byte: 7 nutzbare Bits → 2^7 = 128 Zeichen
- 2 Bytes: 5+6 = 11 nutzbare Bits → 2^{11} = 2 048 Zeichen
- 3 Bytes: 4+6+6 = 16 nutzbare Bits → 2^{16} = 65 536 Zeichen
- 4 Bytes: 3+6+6+6 = 21 nutzbare Bits → 2^{21} = 2 097 152 Zeichen

---

[1)]

Warum?