Studienwoche 2022

Kodierung

08. Juni 2023

ubique

Studienwoche 2023

Kodierung

1. Warum Kodierung
2. ASCII / ISO-8859-1 (Latin 1)
3. UTF-8
4. Base64

Studienwoche 2023

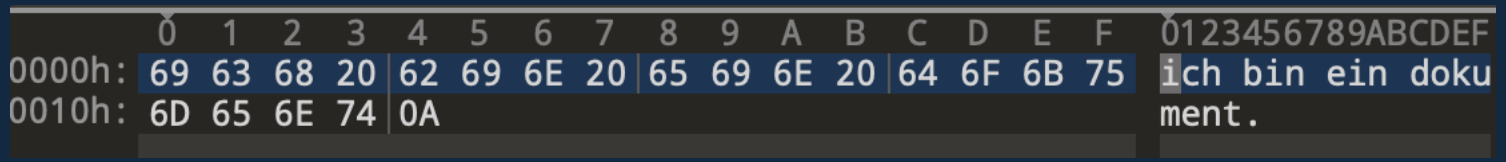
Warum Kodierung

ubique

Studienwoche 2023

Warum Kodierung

- Der Computer kennt kein Konzept von Buchstaben
- Alles was der Computer bearbeiten kann, sind Binärzahlen (resp. Strom)
- Damit wir Text eingeben und lesen können braucht es eine Konvention
- Jeder der meine Datei aufmacht, muss die Zahlen gleich Interpretieren
- Solche Konventionen werden Charset genannt



Studienwoche 2023

ASCII / ISO-8859-1

ubique

ASCII / ISO-8859-1

- ASCII ist die früher im Amerikanischen Sprachraum benutzte Kodierung
- Es gibt die Latin-1 Extensions, welche die Akzente die in Deutsch/Französisch und anderen West-Europäischen Sprachen vorkommen, kodieren
- Kodiert werden alle möglichen Zeichen mit maximal einem Byte (8 bits)
- Es sind also maximal 255 verschiedene Zeichen möglich
- Dies reicht natürlich nicht für alle Sprachen (Kyrillisch, Arabisch, Türkisch usw.)

Studienwoche 2023

ASCII / ISO-8859-1

34	22	"	98	62	b	162	A2	¢	226	E2	â
35	23	#	99	63	c	163	A3	£	227	E3	ã
36	24	$	100	64	d	164	A4	¤	228	E4	ä
37	25	%	101	65	e	165	A5	¥	229	E5	å
38	26	&	102	66	f	166	A6	¦	230	E6	æ
39	27	'	103	67	g	167	A7	§	231	E7	ç
40	28	(	104	68	h	168	A8	¨	232	E8	è
41	29	)	105	69	i	169	A9	©	233	E9	é
42	2A	*	106	6A	j	170	AA	ª	234	EA	ê
43	2B	+	107	6B	k	171	AB	«	235	EB	ë
44	2C	,	108	6C	l	172	AC	¬	236	EC	ì
45	2D	-	109	6D	m	173	AD		237	ED	í
46	2E	.	110	6E	n	174	AE	®	238	EE	î
47	2F	/	111	6F	o	175	AF	¯	239	EF	ï
48	30	0	112	70	p	176	B0	°	240	F0	ð
49	31	1	113	71	q	177	B1	±	241	F1	ñ
50	32	2	114	72	r	178	B2	²	242	F2	ò
51	33	3	115	73	s	179	B3	³	243	F3	ó

Studienwoche 2023

UTF-8

ubique

Studienwoche 2023

UTF-8

- Sprachen-Abhängige Kodierung funktionieren, solange man in einem Land bleibt
- Durch die Globalisierung und die rasante Ausbreitung des Internets führte das zu Problemen
- Texte wurden plötzlich falsch dargestellt (falsche Kodierung)
- Entsprechend wurde eine Kodierung entwickelt, die eine variable Länge an Bytes benutzen kann
- UTF-8 ist in den ersten 128 Zeichen identisch mit ASCII

Studienwoche 2023

UTF-8

- UTF-8 kann bis zu 4 Bytes benutzen (also ca. 2 Mio Zeichen darstellen)
- Dadurch sind 😊, 漢字 und weitere Zeichen in der Gleichen Kodierung möglich

- Fängt das Byte mit 0b0xxxxxxx so ist es Teil von Ascii ($2^7 = 128$)

- Fängt das Byte mit 0b110xxxxx an, so ist es das erste Byte eines Multi-Byte-Zeichen
- Folge Bytes eines Multi-Byte-Zeichen fangen mit 0b10xxxxxx an

- Die Anzahl der 1-Bits vor der ersten 0 im Start-Byte sind die Gesamtzahl der Bytes für dieses Zeichen

- 0b1110xxxx 0b10xxxxxx 0b10xxxxxx

Studienwoche 2023

Base64

ubique

Studienwoche 2023

Base64

- Das Umgekehrte Problem stellt die Darstellung beliebiger Bytes als Alphanumerische Zeichenketten dar

- Dies ist Wertvoll, wenn Protokolle benutzt werden, wo z.B. nur Alphanumerische Zeichen erlaubt sind

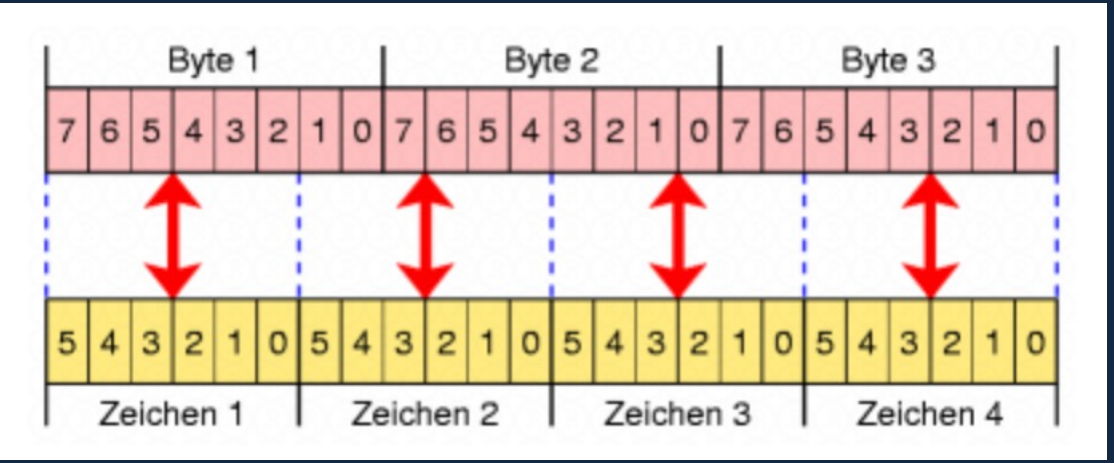
- Will man z.B. ein Bild im HTML-Quellcode direkt hinterlegen, braucht man ein Verfahren, die Bytes (als Schriftzeichen) zu Kodieren

iVBORw0KGgoAAAANSUhEUgAAAUQAAAFECAMAAABoNLf0AAAABGdBTUEAALGPC/xhBQA[...]

Studienwoche 2023

Base64

- Base64 benutzt ein Alphabet aus 64 Zeichen
- Buchstaben A-Z, a-z, 0-9 und _ und –
- Drei Bytes (3*8 = 24) werden zu 4 * 6bit Blöcken ($2^6 = 64$) aufgeteilt
- Enstprechend werden 4/3 Bytes gebraucht um den Gleichen Inhalt zu kodieren



Studienwoche 2023

BigIntegers

ubique

Studienwoche 2023

BigInt

- Für grosse Moduli in RSA brauchen wir sehr grosse Zahlen
- Das normale Limit sind 32bit resp. 64bit Zahlen
- Entsprechend gibt es sogenannte BigIntegers
- Diese Kodieren eine beliebig grosse Zahl
- Dies geschieht, in dem die Bits in eine Liste von Bytes kodiert werden
- Das höchste Bit gibt jeweils das Vorzeichen an (1 => -1, 0 => +1)