



TRIONA – INFORMATION UND TECHNOLOGIE GMBH

Unicode 5.1

Igor Chemnitz Mai 2008

ÜBERSICHT



- **THEORIE**
- **AUSWIRKUNGEN AUF DEN
JAVA-ENTWICKLER**
- **LIVE-DEMO**

THEORIE

- ZIEL VON UNICODE
- WOZU CHARACTER ENCODING ?
- WAS IST UTF ?
- UNICODE – WEIT MEHR ALS NUR CHARACTER ENCODING

ZIEL VON UNICODE

☐ VEREINHEITLICHUNG

.. DER VIELEN HUNDERT VARIANTEN, ZEICHEN ZU CODIEREN

PROBLEME:

- AUSTAUSCH VON DATEIEN / DATEN
- INTERNATIONALISIERUNG IN PROGRAMMEN
- MEHRSPRACHIGKEIT IM SELBEN TEXT
- NICHTALPHABETISCHE ZEICHEN
Z.B FORMELN (MATHEMATIK / CHEMIE /PHYSIK ETC)

☐ 1988 UNICODE CONSORTIUM

non-profit organization

- FIRMEN:** ADOBE, MICROSOFT, IBM, DENIC, GOOGLE, ...
INSTITUTIONEN: INDIA MINISTRY OF IT, UNVER. AT BERKELEY, STAND. ORGS
PRIVATPERSONEN: PAUL DEUTER ...

WOZU

CHARACTER ENCODING?



HALLO



72,65,76,76,79



CHARACTER ENCODING



ASCII-Codetabelle

+	0	1	2	3	4	5	6	7	8	9
30			!	"	#	\$	%	&	'	
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			

CHARACTER ENCODING

z.B. MS-DOS / WINDOWS:



Codepage 850																	Windows 1252																
!	"	#	\$	%	&	'	()	*	+	,	-	.	/	!	"	#	\$	%	&	'	()	*	+	,	-	.	/				
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?		
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O		
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_		
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o		
p	q	r	s	t	u	v	w	x	y	z	{		}	~	p	q	r	s	t	u	v	w	x	y	z	{		}	~				
Ç	ù	é	â	ä	à	ã	ç	ê	ë	è	ï	î	ï	Ä	Å	€	,	f	"	...	+	≠	^	%	Š	≤	Æ	Ž					
É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	x	f	,	'	"	"	•	-	-	~	™	š	≥	æ	ž	ÿ				
á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»	¡	¢	£	¥	¦	§	¨	©	ª	«	¬	-	®	¯				
☄	☄	☄		†	‡	§	¶	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿		
L	⊥	T	†	-	+	ã	Ã	ℒ	ℝ	ℤ	⊥	⊥	=	⊥	x	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï		
ð	Ð	Ê	Ë	È	Í	Î	Ï	J	ŕ	■	■		ï	■	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß			
Ó	Ô	Õ	Ö	Ø	Ù	Ú	Û	Ü	Ý	ˆ	ˆ	ˆ	ˆ	ˆ	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï			
±	=	¾	¶	§	÷	¸	°	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ			

ASCII-Teil der Codetabelle

ASCII-Teil der Codetabelle

Erweiterungen von Codepage 850

Erweiterungen von Windows 1252

CHARACTER ENCODING



Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.iana.org/assignments/character-sets

http://www.iana.org/assignments/character-sets

Character Encoding

- Auto-Detect
- More Encodings
 - Western (ISO-8859-1)
 - Unicode (UTF-8)
 - Western (ISO-8859-15)
 - English (US-ASCII)
 - Western (Windows-1252)
- Customize List...

Reference

- West European
 - East European
 - Baltic (ISO-8859-4)
 - Baltic (ISO-8859-13)
 - Baltic (Windows-1257)
 - Central European (IBM-852)
 - Central European (ISO-8859-2)
 - Central European (MacCE)
 - Central European (Windows-1250)
 - Croatian (MacCroatian)
 - Cyrillic (IBM-855)
 - Cyrillic (ISO-8859-5)
 - Cyrillic (ISO-IR-111)
 - Cyrillic (KOI8-R)
 - Cyrillic (MacCyrillic)
 - Cyrillic (Windows-1251)
 - Cyrillic/Russian (CP-866)
 - Cyrillic/Ukrainian (KOI8-U)
 - Cyrillic/Ukrainian (MacUkrainian)
 - Romanian (ISO-8859-16)
 - Romanian (MacRomanian)
 - East Asian
 - SE & SW Asian
 - Middle Eastern
 - Unicode
 - Unicode (UTF-16)
 - User Defined

Name: ISO_8859-1:1987
MIBenum: 4
Source: ECMA registry
Alias: iso-ir-100
Alias: ISO_8859-1
Alias: ISO-8859-1 (preferred MIME name)
Alias: latin1
Alias: l1
Alias: IBM819
Alias: CP819
Alias: csISOLatin1

SCHRIFTARTEN / FONTS

Byte-Wert: 252

So ist es im Computer gespeichert

Zeichenkodierung

z.B. ISO 8859-1
252 = ü

Schriftart

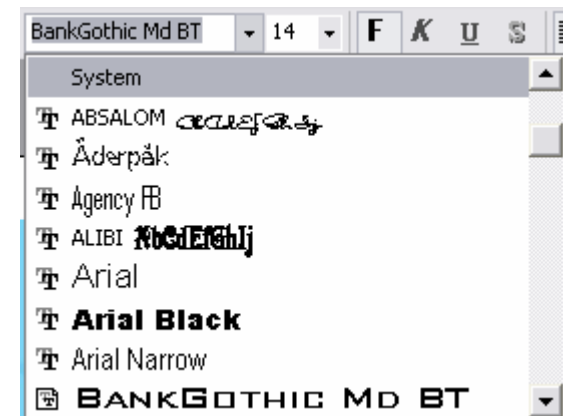
realisiert die Zeichen des Zeichenvorrats

Schriftart "Garamond Italic" stellt "ü" so dar:

ü

iso-8859-1

+	0	1	2	3	4	5	6	7	8	9
160		ı	φ	£	κ	¥	ı	š	"	ø
170	≡	«	¬	-	ø	-	°	±	²	³
180	˘	μ	¶	•	˘	ı	º	»	¼	½
190	¸	ı	À	Á	Â	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
210	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
230	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù
250	ú	û	ü	ý	þ	ÿ				



UNICODE

- 1.114.112 CODEPOINTS (127 IN ASCII)
- 5.0 DEFINIERT JEDOCH „NUR“ 99.024 ZEICHEN, ENTHÄLT
 - DIE ALLERMEISTEN DERZEITIGEN SPRACHEN
LATEINISCHE, ARABISCHE, FERNÖSTLICHE USW.
 - HISTORISCHE SPRACHEN: ABORIGINES, RUNEN
 - MATHEM. SYMBOLE
 - BRAILLE
 - MUSIKALISCHE SYMBOLE
 - ...
- CA 135.000 PRIVATE USE
- DEFINIERT KEINE LOGOS,
KEINE GRAPHIKEN
- UNICODE IST EIN „BUCH“ / EINE NORM,
KEIN TOOL, KEINE SOFTWARE!

Principle	Statement	Principle	Statement
Uniqueness: single, universal and process.	The Unicode Standard defines Unicode text as simple text.	Uniqueness: single, universal and process.	The Unicode Standard defines Unicode text as simple text.
Character semantics: characters have well-defined semantics.	Characters have well-defined semantics.	Character semantics: characters have well-defined semantics.	Characters have well-defined semantics.
Character order: character order is logical or arbitrary.	The default for memory representation is logical order.	Character order: character order is logical or arbitrary.	The default for memory representation is logical order.
Character duplication: duplicate characters are allowed across languages.	The Unicode Standard allows duplicate characters across languages.	Character duplication: duplicate characters are allowed across languages.	The Unicode Standard allows duplicate characters across languages.
Character immutability: characters cannot be reassigned and are immutable.	Character codes are not reassigned and are immutable.	Character immutability: characters cannot be reassigned and are immutable.	Character codes are not reassigned and are immutable.
Compatibility: compatibility between the Unicode Standard and other widely accepted standards.	Unicode Standard and other widely accepted standards.	Compatibility: compatibility between the Unicode Standard and other widely accepted standards.	Unicode Standard and other widely accepted standards.

CHARACTERS NOT GLYPHS

Unicode Characters

U+0041 LATIN CAPITAL LETTER A

U+0061 LATIN SMALL LETTER A

U+043F CYRILLIC SMALL LETTER PE

U+0647 ARABIC LETTER HEH

Glyphs

A A A A A A A A

a a a a a a a a

П п ū

ه ه ه ه

Code Chart:

Code Point

0750

Arabic Supplement

077F

	075	076	077
0	ي 0750	ف 0760	
1	ش 0751	ف 0761	
2	پ 0752	س 0762	

Extended Arabic letters

These are primarily used in Arabic-script orthographies of African languages.

- 0750 ٲ ARABIC LETTER BEH WITH THREE DOTS HORIZONTALLY BELOW
- 0751 ٳ ARABIC LETTER BEH WITH DOT BELOW AND THREE DOTS ABOVE
- 0752 ٴ ARABIC LETTER BEH WITH THREE DOTS POINTING UPWARDS BELOW
- 0753 ٵ ARABIC LETTER BEH WITH THREE DOTS POINTING UPWARDS BELOW AND TWO DOTS ABOVE
- 0754 ٶ ARABIC LETTER BEH WITH TWO DOTS BELOW AND DOT ABOVE
- 0755 ٷ ARABIC LETTER BEH WITH INVERTED SMALL V BELOW

ENCODING FORMS

- **CODE POINT = INTEGER VALUE**
- **1.1 1 4.1 1 2 CODEPOINTS -> MIN 21 BITS -> MIN 3 BYTES**
- **BLOCKWEISE VERTEILUNG AUF 32 BITS (4 BYTES)**

H A L L O

00 00 00 48

00 00 00 41

00 00 00 4C

00 00 00 4C

00 00 00 4F

UTF – UNICODE TRANSFORMATION FORMAT

ALLE FORMEN KODIEREN VOLLSTÄNDIGEN ZEICHENSATZ

VERLUSTFREIE UMWANDLUNG ZW. UTF-32/16/8

WELCHES IST BESSER /SCHLECHTER ? UNTERSCHIEDEN SICH NUR HINSICHTLICH:

- PLATZBEDARF AUF SPEICHERMEDIEN (SPEICHEREFFIZIENZ),
- KODIERUNGS- UND DEKODIERUNGS-AUFWAND (LAUFZEITVERHALTEN)
- KOMPATIBILITÄT ZU ANDEREN (ÄLTEREN) KODIERUNGSARTEN, ZUM BEISPIEL ASCII

A	Ω	語	卍
00000041	000003A9	00008A9E	00010384

UTF-32

tatsächlicher Codepoint-Wert

fixed width

A	Ω	語	卍
0041	03A9	8A9E	D800 DF84

UTF-16

variable width

A	Ω	語	卍
41	CE A9	E8 AA 9E	F0 90 8E 84

UTF-8

variable width

Sonderformen: UTF-7, UTF-EBCDIC ...

UTF-32

A	Ω	語	Ⅲ
00000041	000003A9	00008A9E	00010384

- REPRÄSENTIERT JEDES ZEICHEN MIT EXAKT 4 BYTES**
 - ZAHLENWERT IST IDENTISCH MIT DEM CODEPOINTWERT
 - TROTZ 4 BYTES „NUR,, 1.114.112 CODEPOINTS WEGEN INTEROPERABILITÄT MIT UTF-16 UND UTF-8
- VORTEILE:**
 - SCHNELL - KEINE RECHNEREI ZUM ERMITTELN DES CODEPOINTWERTES
 - NAVIGIEREN DURCH ZEICHENKETTE MIT EINFACHSTER ZEIGER-ARITHMETIK
- NACHTEILE:**
 - VIERFACHER SPEICHERPLATZ IM VERGLEICH ZU ASCII
- EMPFEHLUNG:**
 - WO VERARBEITUNGSGESCHWINDIGKEIT WICHTIGER ALS SPEICHERPLATZ
 - BEI APIS DIE MIT EINZELNEN ZEICHEN ARBEITEN
 - BEI SPRACHEN, DIE SOWIESO MIND. 3 BYTES BENÖTIGEN (JAPANISCH)

UTF-16

A	Ω	語	III
0041	03A9	8A9E	D800 DF84

- **U+0000 BIS U+FFFF IN EINER 16 BIT CODEUNIT**
- **U+10000 BIS U+10FFFF MIT ZWEI 16 BIT CODEUNITS**
- **VORTEILE: - OPTIMIERT FÜR BMP**
BASIC MULTILINGUAL PLANE U+0000 .. U+FFFF
DIE ALLERMEISTEN SCHRIFTZEICHEN SIND IN DER ERSTEN
CODEUNIT ENHALTEN
- **NACHTEILE: - PRÜFUNG AUF ZEICHEN HÖHER ALS U+FFFF:**
Bitfolge 110110 IM ERSTEN 16-BIT-WORT
Bitfolge 110111 IM ZWEITEN 16-BIT-WORT
DANACH GGF. ERRECHNUNG CODEPOINTWERT
- **EMPFEHLUNG: - OPTIMAL FÜR DIE MEISTEN ANWENDUNGSSYSTEME:**
GUTE BALANCE ZW. SPEICHERPLATZ UND
VERARBEITUNGSGESCHWINDIGKEIT

UTF-8

A	Ω	語	III
41	CE A9	E8 AA 9E	F0 90 8E 84

- SPEZIELL KODIERTE BYTE-KETTE MIT VARIABLER LÄNGE:
1, 2, 3 ODER 4 BYTES**
- VORTEILE:**
 - KOMPATIBEL MIT ASCII
 - BYTE-ORIENTIERT, VIELE SYSTEME ARBEITEN MIT SEQUENZEN VON BYTES
 - SPEICHEREFFIZIENT ASSER BEI OST-ASIATISCHEN SPRACHEN
- NACHTEILE:**
 - CODEPOINTWERT MUSS ERRECHNET WERDEN
 - BYTEWERT=CODEPOINT-WERT NUR BEI ASCII-ZEICHEN
- EMPFEHLUNG: FÜR MÖGLICHST KOMPAKTE DATENÜBERTRAGUNG
INTERNET, HTTP, HTML BZW. SPEICHERUNG**

VERGLEICH

ISO-8559-1 vs. UTF-8

Zeichen ä	ISO-8559-1	UTF-8
numerischer Wert in Codetabelle	228	228
Byte-Wert	[228] 1 byte	[195]+[164] 2 bytes

Table 3-6. UTF-8 Bit Distribution

Scalar Value	First Byte	Second Byte	Third Byte	Fourth Byte
00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
000uuuuu zzzyyyy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

ENCODING SCHEMES

DEFINIERT BYTE ORDER

A	Ω	語	Ⅲ
00 00 00 41	00 00 03 A9	00 00 8A 9E	00 01 03 84

UTF-32BE

A	Ω	語	Ⅲ
41 00 00 00	A9 03 00 00	9E 8A 00 00	84 03 01 00

UTF-32LE

**RELEVANT NUR BEI
SERIALISIERTEN DATEN,
DIE BYTEWEISE EINGELESEN
WERDEN
FILES/STREAMS, HTML**

A	Ω	語	Ⅲ
00 41	03 A9	8A 9E	D8 00 DF 84

UTF-16BE

A	Ω	語	Ⅲ
41 00	A9 03	9E 8A	00 D8 84 DF

UTF-16LE

A	Ω	語	Ⅲ
41 CE A9	E8 AA 9E	F0 90 8E 84	

UTF-8

WEIT MEHR ALS NUR CHARACTER ENCODING

UNICODE STANDARD ENTHÄLT REGELN FÜR

- FORMEN VON WÖRTERN
- ZEILENUMBRÜCHE
- SORTIEREN VON TEXT
- FORMATIERUNG VON NUMMERN, DATUM
- SCHREIBEN VON RECHTS NACH LINKS
- ... VIELES MEHR

unzählige Forschungsergebnisse
der weltweiten Sprachwissenschaft



UNICODE & JAVA

DATA TYPE „CHAR“ 16 BIT UNICODE (U+0000 - U+FFFF) UTF-16

```
String s = „hall\u00f6chen“;  
String Gruß = "Hallo, verrückte Welt!";  
System.out.println("Deutsche Umlaute äöü; Unicode-Sonderzeichen: ຈຸດສະຫຼາດ ຈຸນຸທາ");
```

```
javac -encoding UTF-8 Test.java  
Ant: <javac encoding=„UTF-8“> ...
```

```
FileReader eingabe = new FileReader(einausgabedatei);  
FileWriter ausgabe = new FileWriter(einausgabedatei); //GEFAHR !
```

Liest bzw. schreibt mit dem **Standard-Encoding** des Betriebssystems auf dem das Programm läuft (Windows CP1252).

```
InputStreamReader isr = new InputStreamReader(InputStream in, String Encoding);  
OutputStreamWriter( OutputStream out, String enc );
```

Liest bzw. schreibt mit dem als String übergebenen Encoding.

```
Test_de.properties // nur ISO-8859-1 Kodierung erlaubt!
```

UNICODE & JAVA

-AB J2SE 5.0 UNTERSTÜTZUNG FÜR CODEPOINTS > U+FFFF
INNERHALB VON **STRINGS**

-„CHAR“: WEITERHIN 16-BIT !

-KLASSE „STRING“: NEUE METHODEN WIE
- `int codePointCount(int start, int end)`
- `int codePointAt(int index)`

```
private String testString = "abcd\u5B66\uD800\uDF30"; //die letzten zwei char-units  
//bilden hier ein Zeichen oberhalb U+FFFF  
  
int charCount = testString.length(); // liefert wie gehabt Anzahl char-units, also 7  
int cpCount = testString.codePointCount(0, charCount); //liefert 6  
System.out.printf("code point count: %d\n", cpCount);
```

UNICODE & JAVA

JSP:

```
<html>
<%@ page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8" %>
<script type="text/javascript src=„message.js“ charset=„UTF-8“></script>
Text mit tollen zeichen: &#9871; oder auch &#x328e;
...
</html>
```

Filter.java:

```
public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)
    throws IOException, ServletException {
    // Set the characterencoding for the request and response streams.
    req.setCharacterEncoding("UTF-8");
    res.setContentType("text/html; charset=UTF-8");

    // Complete (continue) the processing chain.
    chain.doFilter(req, res);
}
```

QUELLEN



„THE UNICODE STANDARD 5.0“



[HTTP://DE.SELFHTML.ORG](http://de.selfhtml.org)



[HTTP://WWW.SCHOENITZER.DE/ENCODING.HTML](http://www.schoenitzer.de/encoding.html)



[HTTP://WWW.UNICODE.ORG](http://www.unicode.org)



<http://java.sun.com/mailers/techtips/corejava/2006/tt0822.html>

DANKE
FÜR DIE AUFMERKSAMKEIT

