Version	Date	Scripts	Blocks	Total Code Points	Assigned Code Points	Unassigned Code Points	Encoded Char.	Private Use Char.	Non char.	Surr. Code Points	Graphic Char.	Format Char.	Control Char.
1.0.0	October 1991	24	57	65,536	12,795	52,741	7,161	5,632	2	0	7,085	2	74
1.0.1	June 1992	25	59	65,536	34,505	31,031	28,359	6,144	2	0	28,283	2	74
1.1	June 1993	24	63	65,536	40,635	24,901	34,233	6,400	2	0	34,151	2	80
2.0	July 1996	25	67	1,114,112	178,500	935,612	38,950	137,468	34	2,048	38,867	18	65
2.1	May 1998	25	67	1,114,112	178,502	935,610	38,952	137,468	34	2,048	38,869	18	65
3.0	September 1999	38	86	1,114,112	188,809	925,303	49,259	137,468	34	2,048	49,168	26	65
3.1	March 2001	41	95	1,114,112	233,787	880,325	94,205	137,468	66	2,048	94,009	131	65
3.2	March 2002	45	107	1,114,112	234,803	879,309	95,221	137,468	66	2,048	95,023	133	65
4.0	April 2003	52	122	1,114,112	236,029	878,083	96,447	137,468	66	2,048	96,243	139	65
4.1	March 2005	59	142	1,114,112	237,302	876,810	97,720	137,468	66	2,048	97,515	140	65
5.0	July 2006	64	151	1,114,112	238,671	875,441	99,089	137,468	66	2,048	98,884	140	65
5.1	April 2008	75	168	1,114,112	240,295	873,817	100,713	137,468	66	2,048	100,507	141	65
5.2	October 2009	90	194	1,114,112	246,943	867,169	107,361	137,468	66	2,048	107,154	142	65
6.0	October 2010	93	206	1,114,112	249,031	865,081	109,449	137,468	66	2,048	109,242	142	65
6.1	January 2012	100	217	1,114,112	249,763	864,349	110,181	137,468	66	2,048	109,975	141	65

Principi Unicode

- Universalità
- Efficienza
- Importanza ai caratteri e non ai glifi
- Semantica
- Plain text
- Ordine logico
- Unificazione
- Composizione dinamica
- Stabilità
- Convertibilità (round trip conversion)

Tipi di code points

- Assigned
  - Graphic
    Sono i normali caratteri visibili (sia nel BMP che fuori dal BMP)
  - Combining
    Caratteri di composizione (con i caratteri “normali”)
  - Format
    Caratteri “invisibili” di formattazione (spazi, line feed, etc..)
  - Control
    Caratteri di controllo: backspace, bell, etc.., difficilmente distinguibili dai caratteri di formattazione
  - Surrogate
    Hanno un significato specifico per la codifica UTF-16, servono a mappare code points fuori dal BMP utilizzati in coppie alto-basso (high-low)
- Private Use
  Possono essere utilizzati dalle applicazioni e dai vendor (mela della Apple, icone di Whatsapp, etc..)
- Noncharacter
  Non deve corrispondere a nessun carattere e non dovrà mai essere utilizzato per rappresentare qualcosa di significativo: se appare in un flusso di dati deve essere trattato come un errore…
- Unassigned (o reserved)
  Potrà essere utilizzato in versioni future dello standard

La notazione U+nnnn

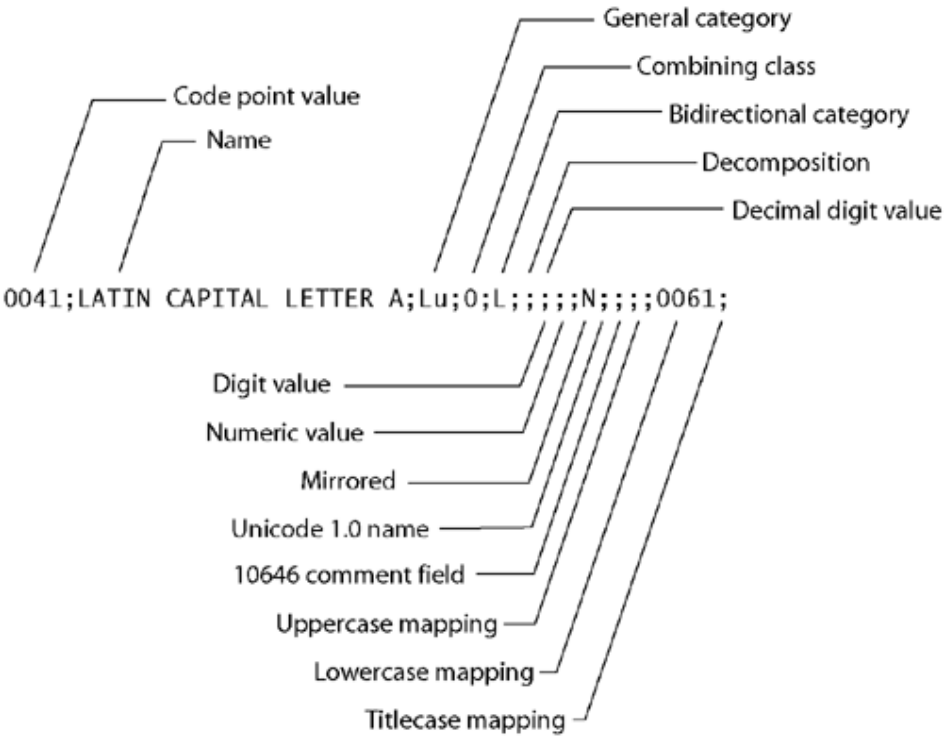
- I caratteri Unicode sono normalmente riferiti con la notazione U+nnnn, dove nnnn è una notazione in cifre esadecimali del valore numerico del code point
- U+0020 ad esempio denota il normale spazio ASCII
- Tale notazione non fa riferimento ad alcuna codifica in particolare
- Usualmente si usano 4 cifre, che bastano a coprire tutto il BMP
- Per caratteri fuori dal BMP si possono usare fino a 6 cifre esadecimali
- Ad esempio:  U+13080

Lo standard Unicode

- Core specifications
- Code charts
- Public Database
- Unicode Tecnical Reports
  - UAX (annessi allo standard)
  - UTS (technical standards): SCSU, Collation
  - UTR
  - Drafts, altro...

Caratteristiche dei caratteri

http://www.unicode.org/Public/UNIDATA/UnicodeData.txt



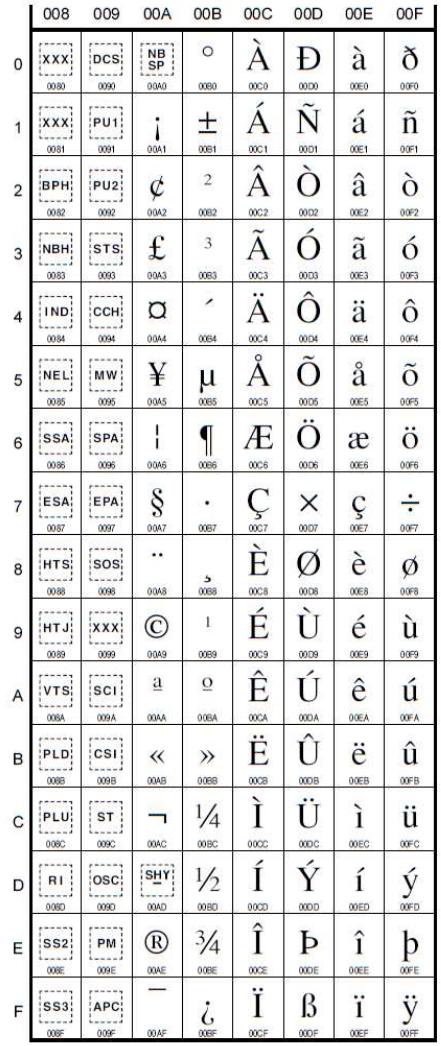
Le caratteristiche dei caratteri sono definite in una serie di database in formato ASCII (☺!!) distribuiti dall'Unicode Consortium

http://www.unicode.org/Public/UNIDATA/

General categories

Code	Description	Sample character
Lu	Letter, uppercase	A
Ll	Letter, lowercase	a
Lt	Letter, titlecase	Dz (U+01C5)
Lm	Letter, modifier	ʰ (U+02B0)
Lo	Letter, other (including ideographs)	א (alef, U+05D0)
Mn	Mark, nonspacing	◌̀ (U+0300)
Mc	Mark, spacing combining	◌ः (U+0903)
Me	Mark, enclosing	(U+06DE)◌۞
Nd	Number, decimal digit	1
Nl	Number, letter	Ⅳ (U+2163)
No	Number, other	½ (U+00BD)
Zs	Separator, space	(space, U+0020)
Zl	Separator, line	(line separator, U+2028)
Zp	Separator, paragraph	(paragraph separator, U+2029)
Cc	Other, control	(carriage return, U+000D)
Cf	Other, format	(soft hyphen, U+00AD)
Cs	Other, surrogate	(surrogate code points)
Co	Other, private use	(U+E000)
Cn	Other, not assigned (including noncharacters)	(U+FFFF, not a character)
Pc	Punctuation, connector	_ (low line, U+005F)
Pd	Punctuation, dash	- (hyphen-minus, U+002D)
Ps	Punctuation, open	(
Pe	Punctuation, close	)
Pi	Punctuation, initial quote	" (U+201C)
Pf	Punctuation, final quote	" (U+201D)
Po	Punctuation, other	!
Sm	Symbol, math	+
Sc	Symbol, currency	$
Sk	Symbol, modifier	^ (circumflex accent, 0+005E)
So	Symbol, other	©

Code charts



C1 Controls and Latin-1 Supplement

	008	009	00A	00B	00C	00D	00E	00F
0	XXX 0080	DCS 0090	NB SP 00A0	° 00B0	À 00C0	Ð 00D0	à 00E0	ð 00F0
1	XXX 0081	PU1 0091	¡ 00A1	± 00B1	Á 00C1	Ñ 00D1	á 00E1	ñ 00F1
2	BPH 0082	PU2 0092	¢ 00A2	2 00B2	Â 00C2	Ò 00D2	â 00E2	ò 00F2
3	NBH 0083	STS 0093	£ 00A3	3 00B3	Ã 00C3	Ó 00D3	ã 00E3	ó 00F3
4	IND 0084	CCH 0094	¤ 00A4	´ 00B4	Ä 00C4	Ô 00D4	ä 00E4	ô 00F4
5	NEL 0085	MW 0095	¥ 00A5	µ 00B5	Å 00C5	Õ 00D5	å 00E5	õ 00F5
6	SSA 0086	SPA 0096	¦ 00A6	¶ 00B6	Æ 00C6	Ö 00D6	æ 00E6	ö 00F6
7	ESA 0087	EPA 0097	§ 00A7	· 00B7	Ç 00C7	× 00D7	ç 00E7	÷ 00F7
8	HTS 0088	SOS 0098	¨ 00A8	¸ 00B8	È 00C8	Ø 00D8	è 00E8	ø 00F8
9	HTJ 0089	XXX 0099	© 00A9	1 00B9	É 00C9	Ù 00D9	é 00E9	ù 00F9
A	VTS 008A	SCI 009A	ª 00AA	º 00BA	Ê 00CA	Ú 00DA	ê 00EA	ú 00FA
B	PLD 008B	CSI 009B	« 00AB	» 00BB	Ë 00CB	Û 00DB	ë 00EB	û 00FB
C	PLU 008C	ST 009C	¬ 00AC	¼ 00BC	Ì 00CC	Ü 00DC	ì 00EC	ü 00FC
D	RI 008D	OSC 009D	SHY 00AD	½ 00BD	Í 00CD	Ý 00DD	í 00ED	ý 00FD
E	SS2 008E	PM 009E	® 00AE	¾ 00BE	Î 00CE	Þ 00DE	î 00EE	þ 00FE
F	SS3 008F	APC 009F	¯ 00AF	¿ 00BF	Ï 00CF	ß 00DF	ï 00EF	ÿ 00FF

00D1 Ñ LATIN CAPITAL LETTER N WITH TILDE
≡ 004E N 0303 ◌̃
00D2 Ò LATIN CAPITAL LETTER O WITH GRAVE
≡ 004F O 0300 ◌̀
00D3 Ó LATIN CAPITAL LETTER O WITH ACUTE
≡ 004F O 0301 ◌́
00D4 Ô LATIN CAPITAL LETTER O WITH CIRCUMFLEX
≡ 004F O 0302 ◌̂
00D5 Õ LATIN CAPITAL LETTER O WITH TILDE
≡ 004F O 0303 ◌̃
00D6 Ö LATIN CAPITAL LETTER O WITH DIAERESIS
≡ 004F O 0308 ◌̈

Mathematical operator

00D7 × MULTIPLICATION SIGN
= z notation Cartesian product
→ 274C ✖ cross mark

Letters

00D8 Ø LATIN CAPITAL LETTER O WITH STROKE
= o slash
→ 2205 ∅ empty set
00D9 Ù LATIN CAPITAL LETTER U WITH GRAVE
≡ 0055 U 0300 ◌̀
00DA Ú LATIN CAPITAL LETTER U WITH ACUTE
≡ 0055 U 0301 ◌́
00DB Û LATIN CAPITAL LETTER U WITH CIRCUMFLEX
≡ 0055 U 0302 ◌̂
00DC Ü LATIN CAPITAL LETTER U WITH DIAERESIS
≡ 0055 U 0308 ◌̈
00DD Ý LATIN CAPITAL LETTER Y WITH ACUTE
≡ 0059 Y 0301 ◌́
00DE Þ LATIN CAPITAL LETTER THORN
00DF ß LATIN SMALL LETTER SHARP S
= Eszett
• German
• uppercase is "SS"
• typographically the glyph for this character can be based on a ligature of 017F ſ with either 0073 s or with an old-style glyph for 007A z (the latter similar in appearance to 0292 ʒ). Both forms exist interchangeably today.
→ 03B2 β greek small letter beta
→ 1E9E ẞ latin capital letter sharp s
00E0 à LATIN SMALL LETTER A WITH GRAVE
≡ 0061 a 0300 ◌̀
00E1 á LATIN SMALL LETTER A WITH ACUTE
≡ 0061 a 0301 ◌́
00E2 â LATIN SMALL LETTER A WITH CIRCUMFLEX
≡ 0061 a 0302 ◌̂
00E3 ã LATIN SMALL LETTER A WITH TILDE
• Portuguese
≡ 0061 a 0303 ◌̃
00E4 ä LATIN SMALL LETTER A WITH DIAERESIS
≡ 0061 a 0308 ◌̈
00E5 å LATIN SMALL LETTER A WITH RING ABOVE
• Danish, Norwegian, Swedish, Walloon
≡ 0061 a 030A ◌̊
00E6 æ LATIN SMALL LETTER AE
= latin small ligature ae (1.0)
= ash (from Old English æsc)
• Danish, Norwegian, Icelandic, Faroese, Old English, French, IPA
→ 0153 œ latin small ligature oe
→ 04D5 ӕ cyrillic small ligature a ie
00E7 ç LATIN SMALL LETTER C WITH CEDILLA
≡ 0063 c 0327 ◌̧
00E8 è LATIN SMALL LETTER E WITH GRAVE
≡ 0065 e 0300 ◌̀
00E9 é LATIN SMALL LETTER E WITH ACUTE
≡ 0065 e 0301 ◌́
00EA ê LATIN SMALL LETTER E WITH CIRCUMFLEX
≡ 0065 e 0302 ◌̂
00EB ë LATIN SMALL LETTER E WITH DIAERESIS
≡ 0065 e 0308 ◌̈
00EC ì LATIN SMALL LETTER I WITH GRAVE
• Italian, Malagasy
≡ 0069 i 0300 ◌̀
00ED í LATIN SMALL LETTER I WITH ACUTE
≡ 0069 i 0301 ◌́
00EE î LATIN SMALL LETTER I WITH CIRCUMFLEX
≡ 0069 i 0302 ◌̂
00EF ï LATIN SMALL LETTER I WITH DIAERESIS
≡ 0069 i 0308 ◌̈
00F0 ð LATIN SMALL LETTER ETH
• Icelandic, Faroese, Old English, IPA
→ 00D0 Ð latin capital letter eth
→ 03B4 δ greek small letter delta
→ 2202 ∂ partial differential
00F1 ñ LATIN SMALL LETTER N WITH TILDE
≡ 006E n 0303 ◌̃
00F2 ò LATIN SMALL LETTER O WITH GRAVE
≡ 006F o 0300 ◌̀
00F3 ó LATIN SMALL LETTER O WITH ACUTE
≡ 006F o 0301 ◌́
00F4 ô LATIN SMALL LETTER O WITH CIRCUMFLEX
≡ 006F o 0302 ◌̂
00F5 õ LATIN SMALL LETTER O WITH TILDE
• Portuguese, Estonian
≡ 006F o 0303 ◌̃
00F6 ö LATIN SMALL LETTER O WITH DIAERESIS
≡ 006F o 0308 ◌̈

Mathematical operator

00F7 ÷ DIVISION SIGN
→ 2215 ∕ division slash
→ 2223 ∣ divides
→ 2797 ➗ heavy division sign

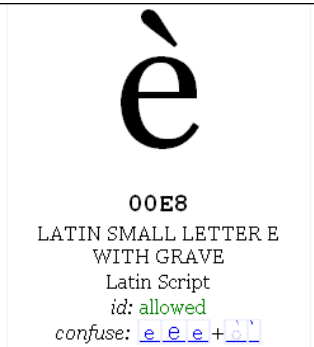
Letters

00F8 ø LATIN SMALL LETTER O WITH STROKE
= o slash
• Danish, Norwegian, Faroese, IPA
00F9 ù LATIN SMALL LETTER U WITH GRAVE
• French, Italian
≡ 0075 u 0300 ◌̀
00FA ú LATIN SMALL LETTER U WITH ACUTE
≡ 0075 u 0301 ◌́
00FB û LATIN SMALL LETTER U WITH CIRCUMFLEX
≡ 0075 u 0302 ◌̂
00FC ü LATIN SMALL LETTER U WITH DIAERESIS
≡ 0075 u 0308 ◌̈

I glifi esemplificativi vengono rilasciati nei "code charts" in PDF (oltre 2000 pagine)

Esistono diversi siti che raccolgono varie informazioni sui caratteri in maniera omogenea

Unicode utilities

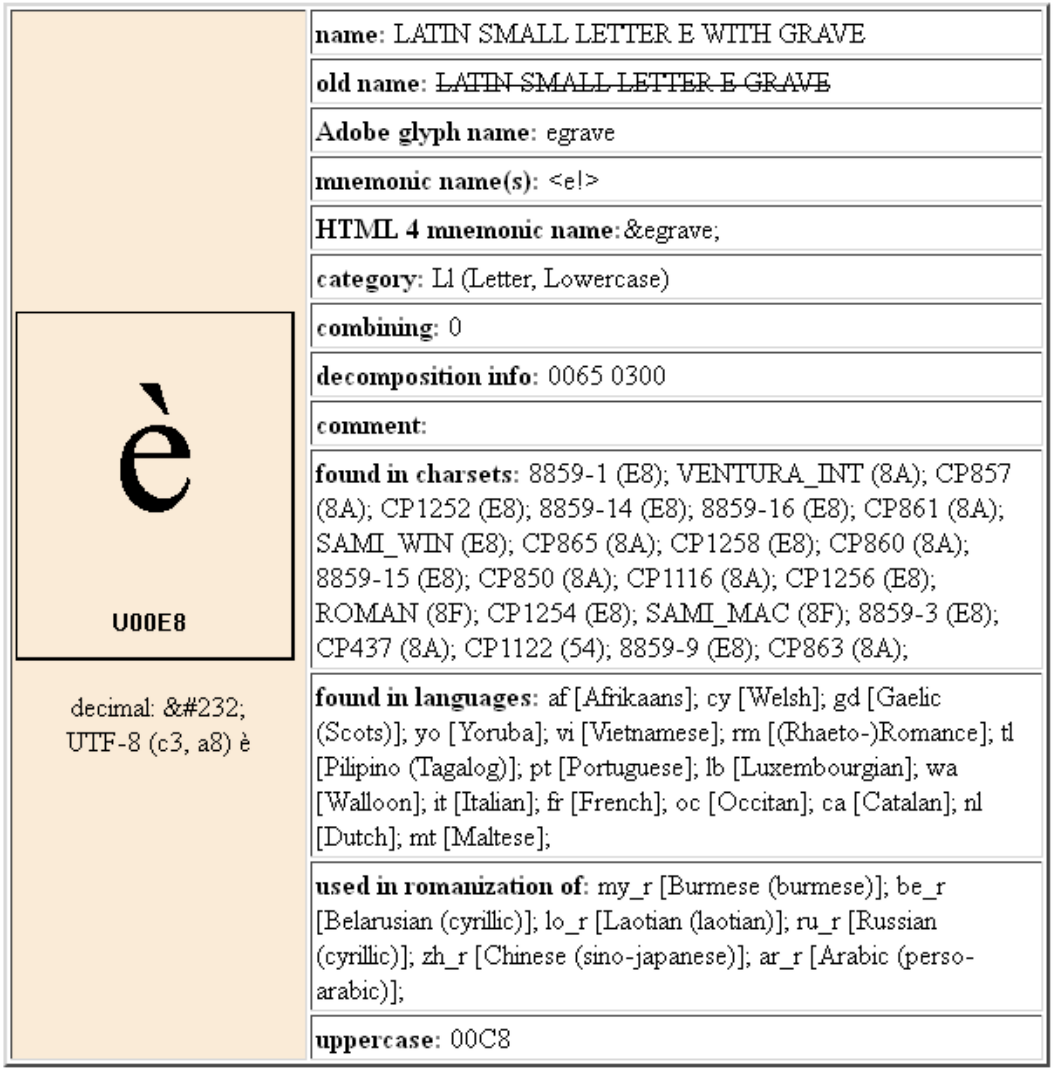


Properties for U+00E8

With Non-Default Values		With Default Values	
Age	1.1	ANY	Yes
alnum	Yes	ASCII	No
Alphabetic	Yes	ASCII_Hex_Digit	No
Block	Latin_1_Supplement	Bidi_Class	Left_To_Right
Case_Sensitive	Yes	Bidi_Control	No
Cased	Yes	Bidi_Mirrored	No
Changes_When_Casemapped	Yes	Bidi_Mirroring_Glyph	è
Changes_When_Titlecased	Yes	blank	No
Changes_When_Uppercased	Yes	bmp	Yes
Decomposition_Type	Canonical	Canonical_Combining_Class	Not_Reordered
East_Asian_Width	Ambiguous	Case_Folding	è
enc_GB2312	A8 A8	Case_Ignorable	No
enc_GBK	A8 A8	Changes_When_Casefolded	No
enc_ISO-8859-1	E8	Changes_When_Lowercased	No
enc_ISO-8859-3	E8	Changes_When_NFKC_Casefolded	No
enc_ISO-8859-9	E8	Dash	No
enc_ISO-8859-15	E8	Default_Ignorable_Code_Point	No
General_Category	Lowercase_Letter	Deprecated	No
graph	Yes	Diacritic	No
Grapheme_Base	Yes	emoji	No
ID_Continue	Yes	enc_Big5	?
ID_Start	Yes	enc_EUC-KR	?
identifier-restriction	recommended	enc_ISO-8859-2	?
idna2003	valid	enc_ISO-8859-4	?
idna2008c	valid	enc_ISO-8859-5	?
is_enc_GB2312	Yes	enc_ISO-8859-6	?
is_enc_GBK	Yes		

http://unicode.org/cldr/utility/character.jsp?a=00E8

Letter Database



è

U00E8

decimal: è
UTF-8 (c3, a8) è

name: LATIN SMALL LETTER E WITH GRAVE

old name: ~~LATIN SMALL LETTER E GRAVE~~

Adobe glyph name: egrave

mnemonic name(s): <e!>

HTML 4 mnemonic name:è

category: Ll (Letter, Lowercase)

combining: 0

decomposition info: 0065 0300

comment:

found in charsets: 8859-1 (E8); VENTURA_INT (8A); CP857 (8A); CP1252 (E8); 8859-14 (E8); 8859-16 (E8); CP861 (8A); SAMI_WIN (E8); CP865 (8A); CP1258 (E8); CP860 (8A); 8859-15 (E8); CP850 (8A); CP1116 (8A); CP1256 (E8); ROMAN (8F); CP1254 (E8); SAMI_MAC (8F); 8859-3 (E8); CP437 (8A); CP1122 (54); 8859-9 (E8); CP863 (8A);

found in languages: af [Afrikaans]; cy [Welsh]; gd [Gaelic (Scots)]; yo [Yoruba]; vi [Vietnamese]; rm [(Rhaeto-)Romance]; tl [Pilipino (Tagalog)]; pt [Portuguese]; lb [Luxembourgian]; wa [Walloon]; it [Italian]; fr [French]; oc [Occitan]; ca [Catalan]; nl [Dutch]; mt [Maltese];

used in romanization of: my_r [Burmese (burmese)]; be_r [Belarusian (cyrillic)]; lo_r [Laotian (laotian)]; ru_r [Russian (cyrillic)]; zh_r [Chinese (sino-japanese)]; ar_r [Arabic (perso-arabic)];

uppercase: 00C8

http://www.eki.ee/letter/chardata.cgi?ucode=00e8

Fileformat.info

Image
è

Browser Test Page | Outline (as SVG file) | Fonts that support U+00E8

Unicode Data	
Name	LATIN SMALL LETTER E WITH GRAVE
Block	Latin-1 Supplement
Category	Letter, Lowercase [Ll]
Combine	0
BIDI	Left-to-Right [L]
Decomposition	LATIN SMALL LETTER E (U+0065) COMBINING GRAVE ACCENT (U+0300)
Mirror	N
Old name	LATIN SMALL LETTER E GRAVE
Index entries	E WITH GRAVE, LATIN SMALL LETTER
Upper case	U+00C8
Title case	U+00C8
Version	Unicode 1.1.0 (June, 1993)

Encodings	
HTML Entity (decimal)	&#232;
HTML Entity (hex)	&#xe8;
HTML Entity (named)	&egrave;
How to type in Microsoft Windows	Alt +E8 Alt 0232 Alt 138
UTF-8 (hex)	0xC3 0xA8 (c3a8)
UTF-8 (binary)	11000011:10101000
UTF-16 (hex)	0x00E8 (00e8)
UTF-16 (decimal)	232
UTF-32 (hex)	0x000000E8 (e8)
UTF-32 (decimal)	232
C/C++/Java source code	"\u00E8"
Python source code	u"\u00E8"

http://www.fileformat.info/info/unicode/char/e8/index.htm

Uniview

U+00E8 LATIN SMALL LETTER E WITH GRAVE

General category:	Ll - Letter, lowercase
Canonical combining class:	0 - Spacing, split, enclosing, reordrant, & Tibetan subjoined
Bidirectional category:	L - Left-to-right
Character decomposition mapping:	0065 0300 è
Unicode 1.0 name:	LATIN SMALL LETTER E GRAVE
Uppercase mapping:	00C8 È
Titlecase mapping:	00C8 È
Decimal:	232
Unicode version:	1.1
As text:	è

More properties at CLDR's Property demo
Descriptions at decodeUnicode
Java data at FileFormat
Use the Conversion tool

Unicode block: Latin-1 Supplement

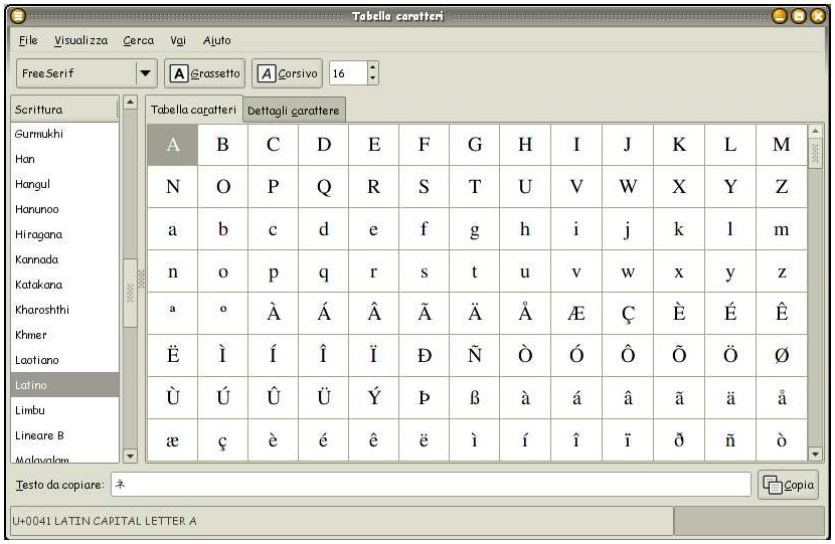
Script group: Letters

Description:
≡ 0065 0300

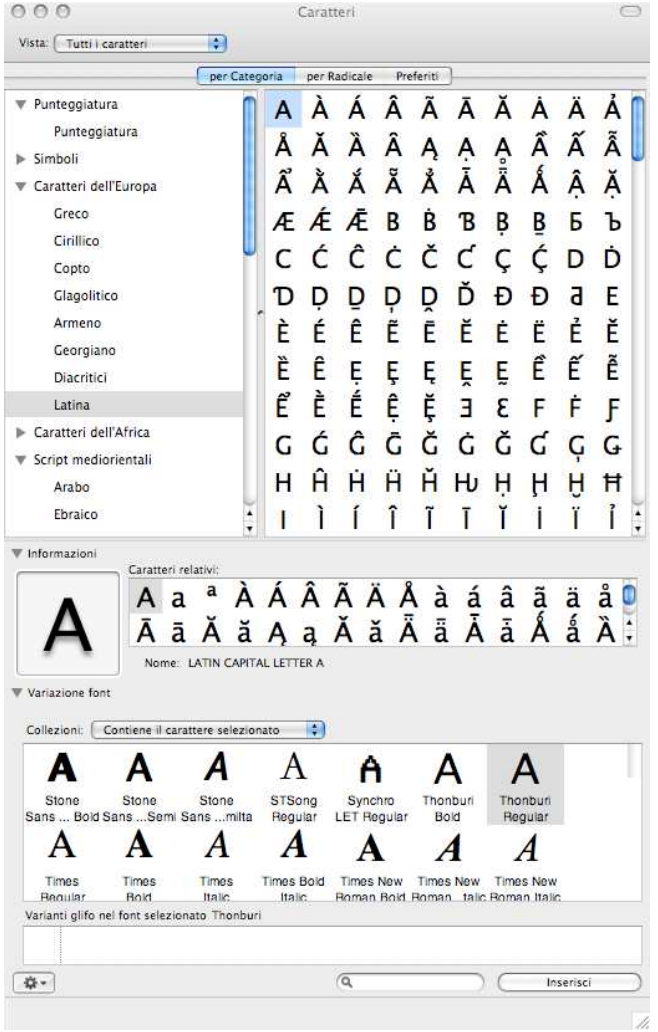
http://rishida.net/scripts/uniview/

Mappe caratteri

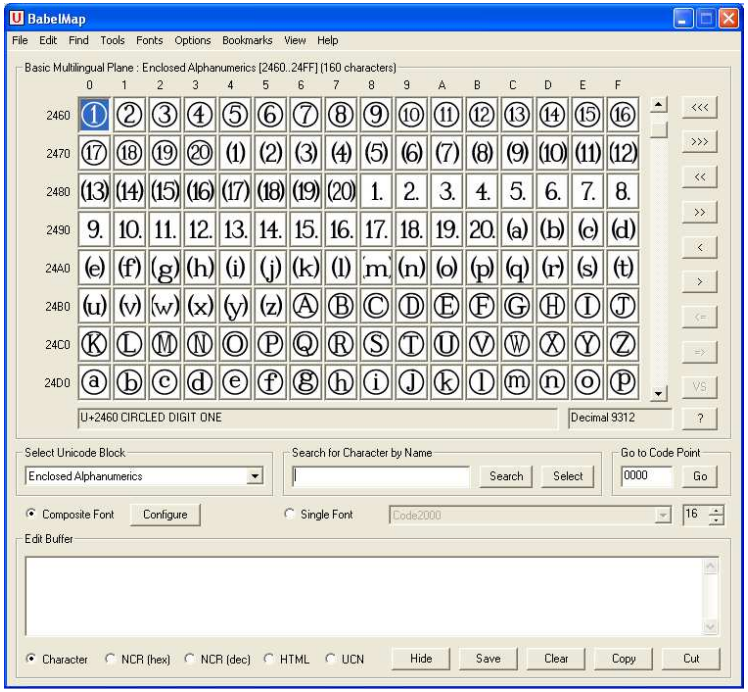
GNOME



MAC



BabelMap



Unicode conformance

- Unicode characters don't fit in 8 bits; deal with it.
- 2 Byte order is only an issue in I/O.
- If you don't know, assume big-endian.
- Loose surrogates have no meaning.
- Neither do U+FFFE and U+FFFF.
- Leave the unassigned codepoints alone.
- It's OK to be ignorant about a character, but not plain wrong.
- Subsets are strictly up to you.
- Canonical equivalence matters.
- Don't garble what you don't understand.
- Process UTF-* by the book.
- Ignore illegal encodings.
- Right-to-left scripts have to go by bidi rules

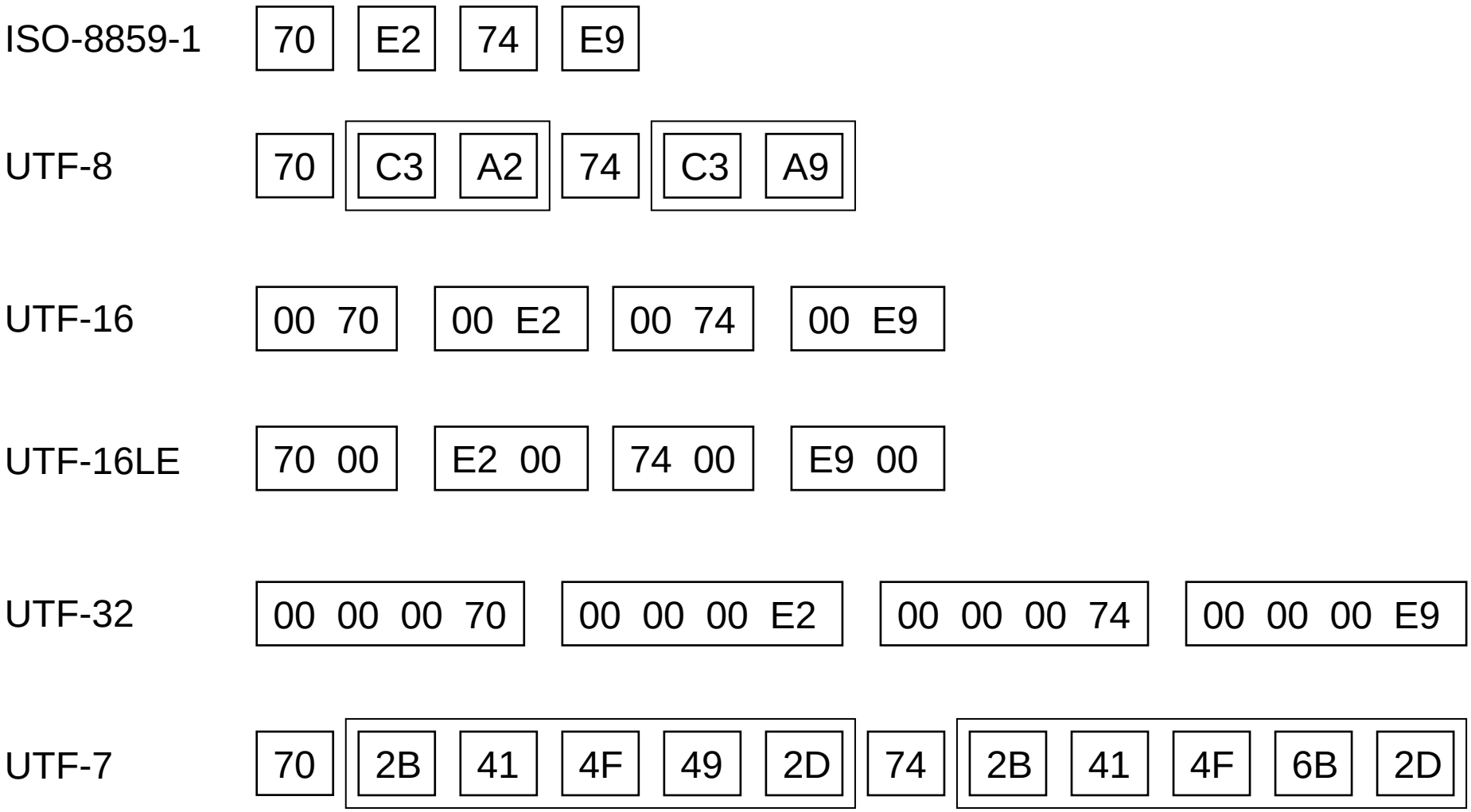
Supporto dei caratteri Unicode

- L’implementazione di Unicode è un processo lungo e graduale
- Anche nelle circostanze in cui Unicode è supportato, tale supporto normalmente non copre tutti i caratteri Unicode (e, soprattutto, non copre tutte le caratteristiche descritte dagli annessi tecnici allo standard)
- Nelle comunicazioni e trasferimento dati è essenziale conoscere quali caratteri il mittente e il destinatario sono in grado di gestire e riconoscere
- Per tale motivo alcuni sottoinsiemi specifici di Unicode sono stati definiti formalmente
- Multilingual Europan Subsets (MES-1, … MES-3)
- Microsoft definisce i suoi Windows Glyph List. Particolarmente importante il WGL4 (Paneuropeo)

Codifiche Unicode

- Unicode definisce tre codifiche (in 5 varianti complessive) per mappare i code point in sequenze di ottetti
- **UTF-32** è la più semplice: utilizza 32 bit, ovvero 4 byte, per ciascun carattere, in maniera estremamente inefficiente dal punto di vista dello spazio, ma molto semplice: ogni sequenza di ottetti ha il valore numerico del code point
- Il valore numerico dipende dalla endianness della codifica (qual è il byte più significativo nella sequenza di 4 byte)
- **UCS-2** è una codifica più vecchia che utilizza 2 byte per ciascun carattere e si limita al BMP (è ISO e non standard Unicode)
- **UTF-16** estende UCS-2, sempre con 2 byte per carattere, ma utilizzando coppie di alcuni particolari code point nel BMP (i *surrogates*) per indirizzare i code point fuori dal BMP secondo un particolare algoritmo
- Anche in UTF-16 occorre tenere conto della endianness
- **UTF-8** è una codifica a numero variabile di byte, progettata per retrocompatibilità con ASCII, senza problemi di endianness e in grado di coprire tutto lo spazio Unicode (e già pronta per una eventuale estensione)
- IETF richiede che tutti i protocolli Internet identifichino l'encoding utilizzato e che ci sia sempre almeno il supporto per UTF-8
- **UTF-7** è una codifica (non standard Unicode) che utilizza solo ottetti con il primo bit sempre a 0 (come se utilizzasse solo 7 bit) mappando i caratteri fuori dal blocco ASCII con sequenze di escape (oltre ai caratteri ASCII utilizzati per iniziare e terminare la stessa sequenza di escape)
- Esistono ulteriori codifiche e varianti (non standard Unicode) di quelle appena viste (**CESU-8, Modified UTF-8, UTF-EBCDIC, SCSU, BOCU**)
- E **UTF-24** che fine ha fatto?

pâté


ISO-8859-1
70 E2 74 E9
UTF-8
70 C3 A2 74 C3 A9
UTF-16
00 70 00 E2 00 74 00 E9
UTF-16LE
70 00 E2 00 74 00 E9 00
UTF-32
00 00 00 70 00 00 00 E2 00 00 00 74 00 00 00 E9
UTF-7
70 2B 41 4F 49 2D 74 2B 41 4F 6B 2D

UTF-32

Carattere		Binary codepoint (21 bit)	Binary UTF-32	Hex UTF-32
M	U+004D	0 0000 0000 0000 0100 1101	00000000 00000000 00000000 01001101	00 00 00 4D
!	U+0021	0 0000 0000 0000 0100 0001	00000000 00000000 00000000 01000001	00 00 00 21
è	U+00E8	0 0000 0000 0000 1110 1000	00000000 00000000 00000000 11101000	00 00 00 E8
€	U+20AC	0 0000 0010 0000 1010 1100	00000000 00000000 00100000 10101100	00 00 20 AC
中	U+4E2D	0 0000 0100 1110 0010 1101	00000000 00000000 01001110 00101101	00 00 4E 2D
	U+13080	0 0001 0011 0000 1000 0000	00000000 00000001 00110000 10000000	00 01 30 80

UTF-16

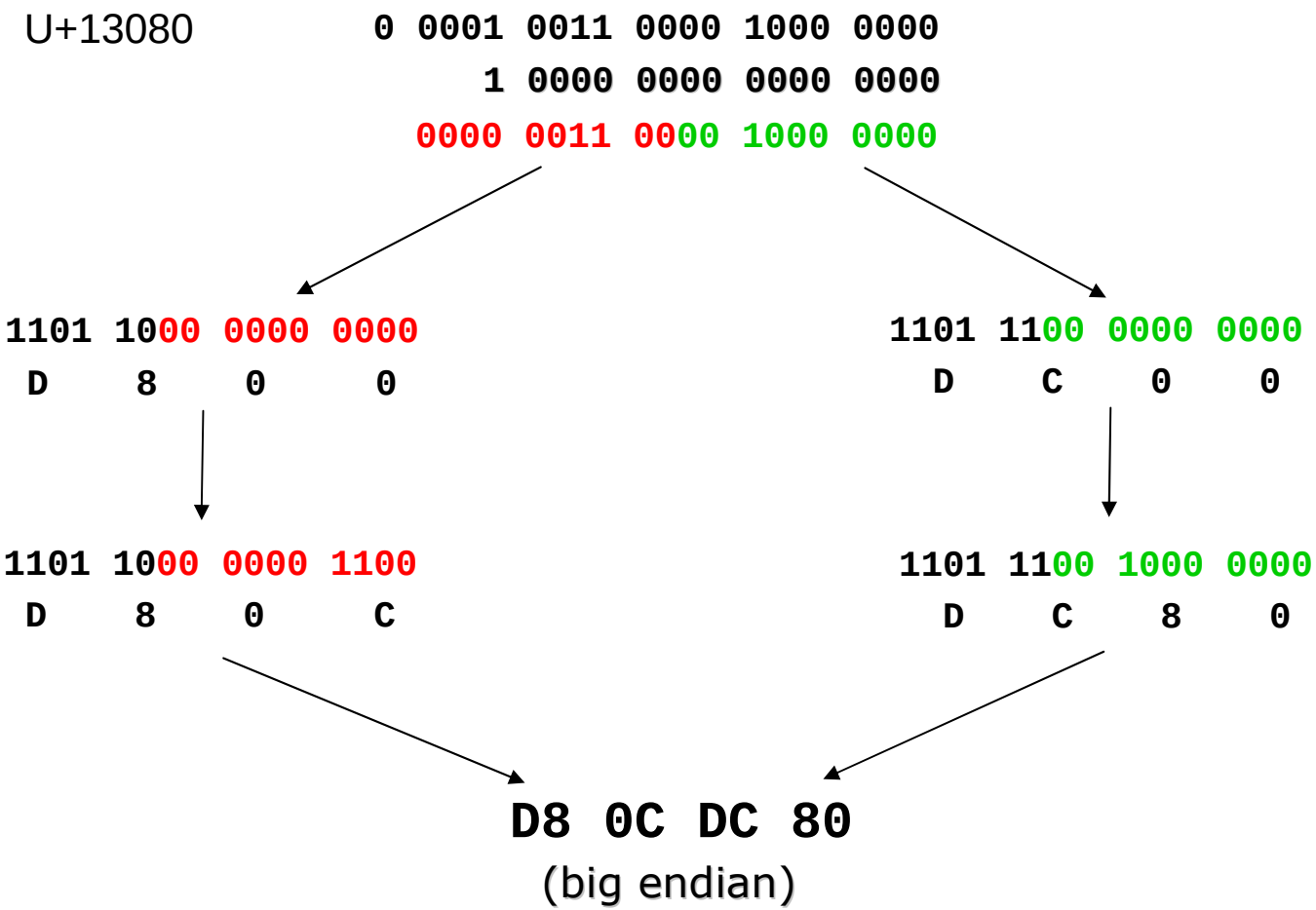
Carattere		Binary codepoint (21 bit)	Binary UTF-16	Hex UTF-16
M	U+004D	0 0000 0000 0000 0100 1101	00000000 01001101	00 4D
!	U+0021	0 0000 0000 0000 0100 0001	00000000 01000001	00 21
è	U+00E8	0 0000 0000 0000 1110 1000	00000000 11101000	00 E8
€	U+20AC	0 0000 0010 0000 1010 1100	00100000 10101100	20 AC
中	U+4E2D	0 0000 0100 1110 0010 1101	01001110 00101101	4E 2D
	U+13080	0 0001 0011 0000 1000 0000	00000000 00000001 00110000 10000000	FAIL!!

UTF-16: U+10000 to U+10FFFF

- Si sottrae 0x10000 dal code point, ottenendo un numero a 20 bit nel range 0..0xFFFFF.
- I primi 10 bit si aggiungono a 0xD800 ottenendo la prima code unit o *lead surrogate*, che sarà nel range 0xD800..0xDBFF (*high surrogates*)
- I restanti 10 bit si aggiungono a 0xDC00 ottenendo la seconda code unit o *trail surrogate*, che sarà nel range 0xDC00..0xDFFF (*low surrogates*)
- Ottengo una coppia di “code points” lead surrogate + trail surrogate che codifico come prima con 2 byte
- I code points da U+D800 to U+DFFF non possono essere usati per altri scopi
- I range per lead surrogates, trail surrogates, e caratteri BMP sono disgiunti, le ricerche sono semplificate
- UCS-2 è una codifica identica a UTF-16, ma senza il meccanismo dei surrogati (codifica solo il BMP)

UTF-16 per 

U+13080



UTF-8

New Jersey - 2 settembre 1992

Ken Thompson

Rob Pike

Bits	Last code point	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	U+007F	0xxxxxxx					
11	U+07FF	110xxxxx	10xxxxxx				
16	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx			
21	U+1FFFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
26	U+3FFFFFF	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+7FFFFFFF	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8: descrizione

Bits	Last code point	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	U+007F	0xxxxxxx					
11	U+07FF	110xxxxx	10xxxxxx				
16	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx			
21	U+1FFFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
26	U+3FFFFFF	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+7FFFFFFF	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

- Per i caratteri del set ASCII utilizza un solo byte, identico a quello che si usa in ASCII (il bit più significativo è sempre a 0)
- Codepoints oltre il 127 sono rappresentati con sequenze multibyte, composte da un byte principale (leading byte) e uno o più byte di continuazione. Il leading byte ha due o più bit high-order a 1; i byte di continuazione cominciano sempre con 10.
- Il numero dei bit a 1 nel leading byte di una sequenza multibyte indica il numero di byte della sequenza stessa: la lunghezza della sequenza può essere determinata senza dover esaminare l'intera sequenza
- Single byte, leading byte e continuation byte non possono mai avere lo stesso valore: lo schema è auto-sincronizzante. Si può trovare l'inizio di un carattere tornando indietro al massimo di 5 byte (3 byte nell'implementazione attuale)

UTF-8: esempi

Carattere		Binary codepoint	Binary UTF-8	Hex UTF-8
M	U+004D *<007F*	00000000 01001101	0XXXXXXX 01001101	4D
!	U+0021 *<007F*	00000000 01000001	0XXXXXXX 01000001	21
è	U+00E8 *0080-07FF*	00000000 11101000	110XXXXX 10XXXXXX 11000011 10101000	C3 A8
€	U+20AC *0800-FFFF*	00100000 10101100	1110XXXX 10XXXXXX 10XXXXXX 11100010 10000010 10101100	E2 82 AC
中	U+4E2D *0800-FFFF*	01001110 00101101	1110XXXX 10XXXXXX 10XXXXXX 11100100 10111000 10101101	E4 B8 AD
	U+13080 *>FFFF*	00000001 00110000 10000000	11110XXX 10XXXXXX 10XXXXXX 10XXXXXX 11110000 10010011 10000010 10000000	F0 93 82 80

UTF-8: vantaggi

- In generale
  - Retrocompatibilità con ASCII (porta con sé molti vantaggi)
  - Può essere riconosciuto con metodi euristici più facilmente di altri encoding
  - Può essere esteso (anche oltre i 31 bit!)

- Rispetto ad altre codifiche single-byte
  - Può rappresentare qualunque carattere Unicode
  - I byte 0xFE e 0xFF non possono mai comparire (BOM UTF-16, Telnet)
- Rispetto ad altre codifiche multi-byte
  - Estensione “propria” ad ASCII come se fosse una single-byte
  - Autosincronizzante
  - Consente l’uso di algoritmi di ricerca stringa byte-oriented
  - Efficienza della codifica (operazioni sui bit)

- Rispetto a UTF-16
  - Nessun problema di endianness (è byte ordered)
  - Più sicuro (per la presenza di sequenze di byte “vietate”)
  - Più facile implementare la retrocompatibilità su determinati vecchi programmi (grazie alla retrocompatibilità con ASCII)
  - Caratteri fuori dal BMP non sono casi “speciali”
  - Dimensioni (in byte totali) generalmente più piccole di UTF-16 (sempre vero con code points < U+0800)

UTF-8: svantaggi

- In generale
  - Un parser UTF-8 non perfettamente compatibile potrebbe consentire un certo tipo di attacchi basati su una codifica multibyte “vietata” equivalente a una sequenza single-byte di un carattere non consentito (attacchi a Web server bacati nel 2001)
  - In mancanza di BOM e altre indicazioni può essere indistinguibile da ASCII e da ISO-8859-1 rendendo impossibile per i programmi determinare automaticamente la codifica e provocando mal di testa nei programmatori
- Rispetto ad altre codifiche single-byte
  - Il testo codificato in UTF-8 richiede in generale più spazio (in byte) rispetto alla codifica più appropriata per determinati script. (Particolari critiche in India: x3)
  - Essendo una codifica multi-byte a lunghezza variabile diventa più difficile determinare la lunghezza in caratteri di una stringa (ed effettuare estrazione di sottostringhe)
- Rispetto ad altre codifiche multi-byte
  - Anche rispetto ad altre codifiche multibyte può richiedere più spazio per determinati script
- Rispetto a UTF-16
  - Stesso problema

UTF-8: codepage layout

UTF-8	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	NUL 0000 *0*	SOH 0001 *1*	STX 0002 *2*	ETX 0003 *3*	EOT 0004 *4*	ENQ 0005 *5*	ACK 0006 *6*	BEL 0007 *7*	BS 0008 *8*	HT 0009 *9*	LF 000A *10*	VT 000B *11*	FF 000C *12*	CR 000D *13*	SO 000E *14*	SI 000F *15*
1_	DLE 0010 *16*	DC1 0011 *17*	DC2 0012 *18*	DC3 0013 *19*	DC4 0014 *20*	NAK 0015 *21*	SYN 0016 *22*	ETB 0017 *23*	CAN 0018 *24*	EM 0019 *25*	SUB 001A *26*	ESC 001B *27*	FS 001C *28*	GS 001D *29*	RS 001E *30*	US 001F *31*
2_	SP 0020 *32*	! 0021 *33*	" 0022 *34*	# 0023 *35*	$ 0024 *36*	% 0025 *37*	& 0026 *38*	' 0027 *39*	(0028 *40*	) 0029 *41*	* 002A *42*	+ 002B *43*	, 002C *44*	- 002D *45*	. 002E *46*	/ 002F *47*
3_	0 0030 *48*	1 0031 *49*	2 0032 *50*	3 0033 *51*	4 0034 *52*	5 0035 *53*	6 0036 *54*	7 0037 *55*	8 0038 *56*	9 0039 *57*	: 003A *58*	; 003B *59*	< 003C *60*	= 003D *61*	> 003E *62*	? 003F *63*
4_	@ 0040 *64*	A 0041 *65*	B 0042 *66*	C 0043 *67*	D 0044 *68*	E 0045 *69*	F 0046 *70*	G 0047 *71*	H 0048 *72*	I 0049 *73*	J 004A *74*	K 004B *75*	L 004C *76*	M 004D *77*	N 004E *78*	O 004F *79*
5_	P 0050 *80*	Q 0051 *81*	R 0052 *82*	S 0053 *83*	T 0054 *84*	U 0055 *85*	V 0056 *86*	W 0057 *87*	X 0058 *88*	Y 0059 *89*	Z 005A *90*	[005B *91*	\ 005C *92*	] 005D *93*	^ 005E *94*	_ 005F *95*
6_	` 0060 *96*	a 0061 *97*	b 0062 *98*	c 0063 *99*	d 0064 *100*	e 0065 *101*	f 0066 *102*	g 0067 *103*	h 0068 *104*	i 0069 *105*	j 006A *106*	k 006B *107*	l 006C *108*	m 006D *109*	n 006E *110*	o 006F *111*
7_	p 0070 *112*	q 0071 *113*	r 0072 *114*	s 0073 *115*	t 0074 *116*	u 0075 *117*	v 0076 *118*	w 0077 *119*	x 0078 *120*	y 0079 *121*	z 007A *122*	{ 007B *123*	| 007C *124*	} 007D *125*	~ 007E *126*	DEL 007F *127*
8_	• +00 *128*	• +01 *129*	• +02 *130*	• +03 *131*	• +04 *132*	• +05 *133*	• +06 *134*	• +07 *135*	• +08 *136*	• +09 *137*	• +0A *138*	• +0B *139*	• +0C *140*	• +0D *141*	• +0E *142*	• +0F *143*
9_	• +10 *144*	• +11 *145*	• +12 *146*	• +13 *147*	• +14 *148*	• +15 *149*	• +16 *150*	• +17 *151*	• +18 *152*	• +19 *153*	• +1A *154*	• +1B *155*	• +1C *156*	• +1D *157*	• +1E *158*	• +1F *159*
A_	• +20 *160*	• +21 *161*	• +22 *162*	• +23 *163*	• +24 *164*	• +25 *165*	• +26 *166*	• +27 *167*	• +28 *168*	• +29 *169*	• +2A *170*	• +2B *171*	• +2C *172*	• +2D *173*	• +2E *174*	• +2F *175*
B_	• +30 *176*	• +31 *177*	• +32 *178*	• +33 *179*	• +34 *180*	• +35 *181*	• +36 *182*	• +37 *183*	• +38 *184*	• +39 *185*	• +3A *186*	• +3B *187*	• +3C *188*	• +3D *189*	• +3E *190*	• +3F *191*
2-byte C_	2-byte inval (0000) *192*	2-byte inval (0040) *193*	Latin-1 0080 *194*	Latin-1 00C0 *195*	Latin Ext-A 0100 *196*	Latin Ext-A 0140 *197*	Latin Ext-B 0180 *198*	Latin Ext-B 01C0 *199*	Latin Ext-B 0200 *200*	IPA 0240 *201*	IPA 0280 *202*	Spaci Modif 02C0 *203*	Combi Diacr 0300 *204*	Combi Diacr 0340 *205*	Greek 0380 *206*	Greek 03C0 *207*
2-byte D_	Cyril 0400 *208*	Cyril 0440 *209*	Cyril 0480 *210*	Cyril 04C0 *211*	Cyril 0500 *212*	Armen 0540 *213*	Hebrew 0580 *214*	Hebrew 05C0 *215*	Arabic 0600 *216*	Arabic 0640 *217*	Arabic 0680 *218*	Arabic 06C0 *219*	Syriac 0700 *220*	Arabic 0740 *221*	Thaana 0780 *222*	N'Ko 07C0 *223*
3-byte E_	Indic 0800* *224*	Misc. 1000 *225*	Symbol 2000 *226*	Kana CJK 3000 *227*	CJK 4000 *228*	CJK 5000 *229*	CJK 6000 *230*	CJK 7000 *231*	CJK 8000 *232*	CJK 9000 *233*	Asian A000 *234*	Hangul B000 *235*	Hangul C000 *236*	Hangul Surr D000 *237*	Priv Use E000 *238*	Forms F000 *239*
4-byte F_	Ancient Sym,CJK 10000* *240*	unall 40000 *241*	unall 80000 *242*	Tags Priv C0000 *243*	Priv Use 100000 *244*	4-byte inval 140000 *245*	4-byte inval 180000 *246*	4-byte inval 1C0000 *247*	5-byte inval 200000* *248*	5-byte inval 3000000 *249*	5-byte inval 3400000 *250*	5-byte inval 3800000 *251*	6-byte inval 4000000* *252*	6-byte inval 4400000 *253*	*254*	*255*

UTF-8: sequenze non valide

- I byte “rossi” nella tavola precedente
- Un byte di continuazione inatteso
- Uno start byte non seguito dal numero previsto di byte di continuazione
- Una sequenza la cui decodifica rappresenta un code point per il quale esiste una sequenza di codifica più corta
- Una sequenza di 4 byte che codifica un code point oltre il U+10FFFF

UTF-8: code points non validi

- I code point “surrogati” (da U+D800 a U+DFFF) **NON** possono essere a loro volta codificati in UTF-8: il loro utilizzo è riservato a UTF-16
- In altre parole, per codificare i code point fuori dal BMP in UTF-8 si deve utilizzare la sequenza prevista UTF-8 (di 4 byte)
- Codificare i code point surrogati in UTF-8 significherebbe applicare una “doppia codifica”