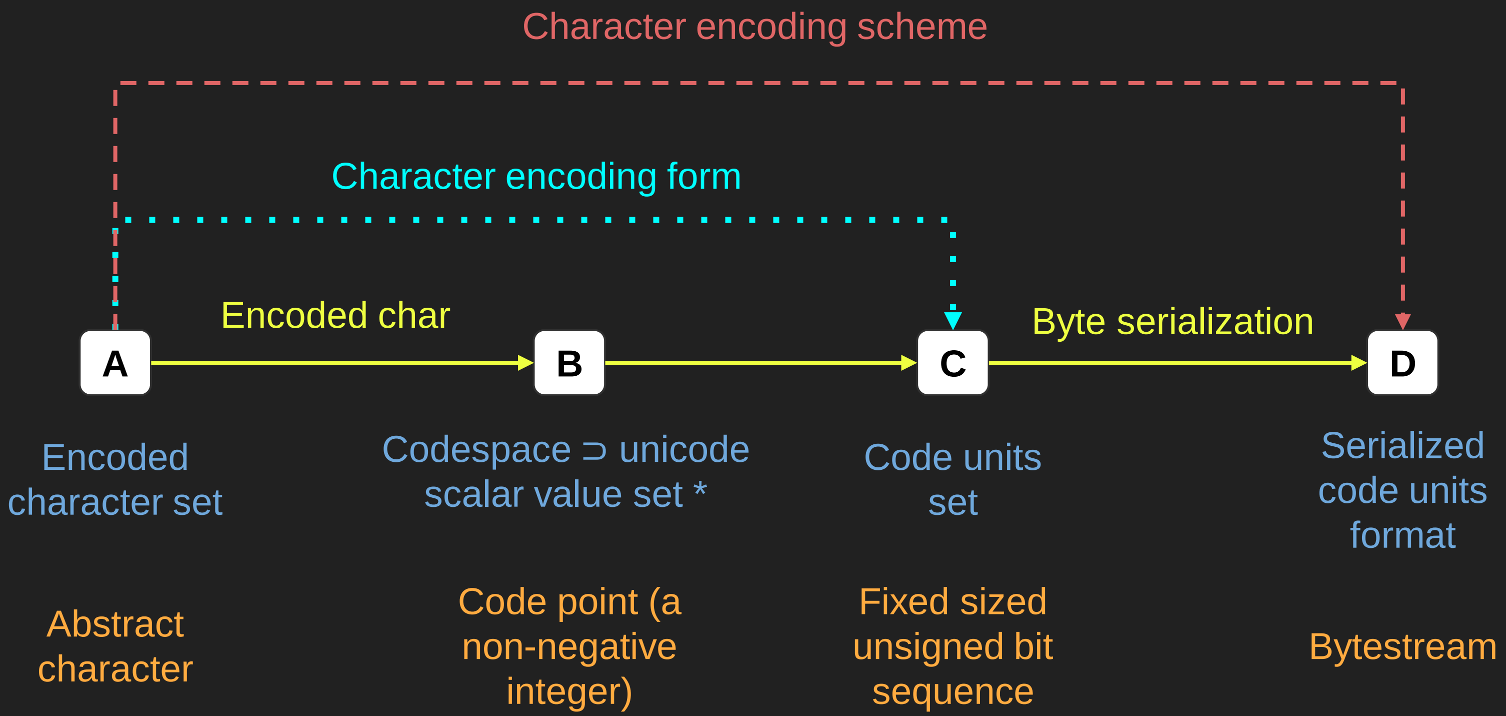
Digital representation of writing systems

Concept	Character (c)	Grapheme (G)	Glyph (H)	Glyph image (h∈H)
Kind of object	Abstract	Abstract	Abstract	Concrete
Minimal unit of	Semantic value (e.g. representation, control, organization)	Distinction value	Form/ Shape	Realization of the shape
Example	**c** = LATIN SMALL LETTER A U+0061	c ∈ **G** := { LATIN SMALL LETTER A U+0061, LATIN CAPITAL LETTER A U+0061}	**H(c)** := { a, a, **a**, *a*, ***a***, ***a***, *a*, … }	a
	“Arbitre” and “arbitre” have the same meaning, does not exist distinction between those two words		This does have more to do with typography than encoding	



Character encoding scheme

Character encoding form

Encoded char

Byte serialization

A

B

C

D

Encoded character set

Codespace ⊃ unicode scalar value set *

Code units set

Serialized code units format

Abstract character

Code point (a non-negative integer)

Fixed sized unsigned bit sequence

Bytestream

(*) Codespace set \ {surrogate code points} = scalar value set

Digital representation of characters

Coded character	Code point	UTF-8 Code units	UTF-16 Code units	UTF-32 Code units
ε	U+03B5	0xCE 0xB5	0x03B5	0x000003B5
	11 1011 0101	1100 1110 1011 0101	0000 0011 1011 0101	0000 0000 0000 0000 0000 0011 1011 0101
𝑣	U+1D463	0xF0 0x9D 0x91 0xA3	0xD835 0xDC63	0x0001D463
	1 1101 0100 0110 0011	1111 0000 1001 1101 1001 0001 1010 0011	1101 1000 0011 0101 1101 1100 0110 0011	0000 0000 0000 0001 1101 0100 0110 0011

Color legend

Blue: binary representation.

Red: most significant bit.

Green: least significant bit.

Gray: Leading zeros to fill the code unit word size.

Yellow: UTF-8. Leading bytes representing number of bytes.

Orange: UTF-16. Leading bytes representing high and low surrogates (only for values bigger than U+FFFF)

Serialization

UTF-8 schema: same order

UTF-16 schema: big-endian or little-endian

UTF-32 schema: big-endian or little-endian