<https://en.wikipedia.org/wiki/Character_encoding>

Terminology[[edit](https://en.wikipedia.org/w/index.php?title=Character_encoding&action=edit&section=2)]

**Terminology related to character encoding**



* A *character* is a minimal unit of text that has semantic value.
* A *character set* is a collection of characters that might be used by multiple languages. *Example:* The Latin character set is used by English and most European languages, though the Greek character set is used only by the Greek language.
* A *coded character set* is a character set in which each character corresponds to a unique number.
* A *code point* of a coded character set is any allowed value in the character set.
* A *code unit* is a bit sequence used to encode each character of a repertoire within a given encoding form.

**Character repertoire (the abstract set of characters)**

The character repertoire is an abstract set of more than one million characters found in a wide variety of scripts including Latin, Cyrillic, Chinese, Korean, Japanese, Hebrew, and Aramaic.

Other symbols such as musical notation are also included in the character repertoire. Both the Unicode and [GB18030](https://en.wikipedia.org/wiki/GB18030) standards have a character repertoire. As new characters are added to one standard, the other standard also adds those characters, to maintain parity.

The code unit size is equivalent to the bit measurement for the particular encoding:

* A code unit in [US-ASCII](https://en.wikipedia.org/wiki/US-ASCII) consists of 7 bits;
* A code unit in [UTF-8](https://en.wikipedia.org/wiki/UTF-8), [EBCDIC](https://en.wikipedia.org/wiki/EBCDIC) and GB18030 consists of 8 bits;
* A code unit in [UTF-16](https://en.wikipedia.org/wiki/UTF-16) consists of 16 bits;
* A code unit in [UTF-32](https://en.wikipedia.org/wiki/UTF-32) consists of 32 bits.

*Example of a code unit:* Consider a [string](https://en.wikipedia.org/wiki/String_%28computer_science%29) of the letters "abc" followed by U+10400 𐐀 deseret capital letter long i (represented with 1 char32\_t, 2 char16\_t or 4 char8\_t). That string contains:

* four characters;
* four code points
* either:

four code units in UTF-32 (00000061, 00000062, 00000063, 00010400)

five code units in UTF-16 (0061, 0062, 0063, d801, dc00), or

seven code units in UTF-8 (61, 62, 63, f0, 90, 90, 80).

The convention to refer to a character in Unicode is to start with 'U+' followed by the codepoint value in hexadecimal. The range of valid code points for the Unicode standard is U+0000 to U+10FFFF, inclusive, divided in 17 [planes](https://en.wikipedia.org/wiki/Plane_%28Unicode%29), identified by the numbers 0 to 16. Characters in the range U+0000 to U+FFFF are in plane 0, called the [Basic Multilingual Plane](https://en.wikipedia.org/wiki/Plane_%28Unicode%29#Basic_Multilingual_Plane) (BMP). This plane contains most commonly-used characters. Characters in the range U+10000 to U+10FFFF in the other planes are called [supplementary characters](https://en.wikipedia.org/wiki/Supplementary_characters).

The following table shows examples of code point values:

|  |  |  |
| --- | --- | --- |
| **Character** | **Unicode code point** | **Glyph** |
| Latin A | U+0041 | Α |
| Latin sharp S | U+00DF | ß |
| Han for East | U+6771 | 東 |
| Ampersand | U+0026 | & |
| Inverted exclamation mark | U+00A1 | ¡ |
| Section sign | U+00A7 | § |

A code point is represented by a sequence of code units. The mapping is defined by the encoding. Thus, the number of code units required to represent a code point depends on the encoding:

* *UTF-8:* code points map to a sequence of one, two, three or four code units.
* *UTF-16:* code units are twice as long as 8-bit code units. Therefore, any code point with a scalar value less than U+10000 are encoded with a single code unit. Code points with a value U+10000 or higher require two code units each. These pairs of code units have a unique term in UTF-16: ["Unicode surrogate pairs".](https://en.wikipedia.org/wiki/UTF-16#cite_note-Unicode7Ch3s8-5)
* *UTF-32:* the 32-bit code unit is large enough that every code point is represented as a single code unit.
* *GB18030:* multiple code units per code point are common, because of the small code units. Code points are mapped to one, two, or four code units.[[4]](https://en.wikipedia.org/wiki/Character_encoding#cite_note-4)

Unicode encoding model[[edit](https://en.wikipedia.org/w/index.php?title=Character_encoding&action=edit&section=3)]

[Unicode](https://en.wikipedia.org/wiki/Unicode) and its parallel standard, the ISO/IEC 10646 [Universal Character Set](https://en.wikipedia.org/wiki/Universal_Character_Set), together constitute a modern, unified character encoding. Rather than mapping characters directly to octets ([bytes](https://en.wikipedia.org/wiki/Byte)), they separately define what characters are available, corresponding natural numbers ([code points](https://en.wikipedia.org/wiki/Code_point)), how those numbers are encoded as a series of fixed-size natural numbers (code units), and finally how those units are encoded as a stream of octets. The purpose of this decomposition is to establish a universal set of characters that can be encoded in a variety of ways.[[5]](https://en.wikipedia.org/wiki/Character_encoding#cite_note-utr17-5) To describe this model correctly requires more precise terms than "character set" and "character encoding." The terms used in the modern model follow:[[5]](https://en.wikipedia.org/wiki/Character_encoding#cite_note-utr17-5)

A **character repertoire** is the full set of abstract characters that a system supports. The repertoire may be closed, i.e. no additions are allowed without creating a new standard (as is the case with ASCII and most of the ISO-8859 series), or it may be open, allowing additions (as is the case with Unicode and to a limited extent the [Windows code pages](https://en.wikipedia.org/wiki/Windows_code_page)). The characters in a given repertoire reflect decisions that have been made about how to divide writing systems into basic information units. The basic variants of the [Latin](https://en.wikipedia.org/wiki/Latin_alphabet), [Greek](https://en.wikipedia.org/wiki/Greek_alphabet) and [Cyrillic](https://en.wikipedia.org/wiki/Cyrillic) alphabets can be broken down into letters, digits, punctuation, and a few *special characters* such as the space, which can all be arranged in simple linear sequences that are displayed in the same order they are read. But even with these alphabets, [diacritics](https://en.wikipedia.org/wiki/Diacritic) pose a complication: they can be regarded either as part of a single character containing a letter and diacritic (known as a precomposed character), or as separate characters. The former allows a far simpler text handling system but the latter allows any letter/diacritic combination to be used in text. [Ligatures](https://en.wikipedia.org/wiki/Typographic_ligature) pose similar problems. Other writing systems, such as Arabic and Hebrew, are represented with more complex character repertoires due to the need to accommodate things like bidirectional text and [glyphs](https://en.wikipedia.org/wiki/Glyph) that are joined together in different ways for different situations.

A **coded character set** (CCS) is a [function](https://en.wikipedia.org/wiki/Function_%28mathematics%29) that maps characters to [*code points*](https://en.wikipedia.org/wiki/Code_point) (each code point represents one character). For example, in a given repertoire, the capital letter "A" in the Latin alphabet might be represented by the code point 65, the character "B" to 66, and so on. Multiple coded character sets may share the same repertoire; for example [ISO/IEC 8859-1](https://en.wikipedia.org/wiki/ISO/IEC_8859-1) and IBM code pages 037 and 500 all cover the same repertoire but map them to different code points.

A **character encoding form** (CEF) is the mapping of code points to *code units* to facilitate storage in a system that represents numbers as bit sequences of fixed length (i.e. practically any computer system). For example, a system that stores numeric information in 16-bit units can only directly represent code points 0 to 65,535 in each unit, but larger code points (say, 65,536 to 1.4 million) could be represented by using multiple 16-bit units. This correspondence is defined by a CEF.

Next, a **character encoding scheme** (CES) is the mapping of code units to a sequence of octets to facilitate storage on an octet-based file system or transmission over an octet-based network. Simple character encoding schemes include [UTF-8](https://en.wikipedia.org/wiki/UTF-8), [UTF-16BE](https://en.wikipedia.org/wiki/UTF-16BE), [UTF-32BE](https://en.wikipedia.org/wiki/UTF-32BE), [UTF-16LE](https://en.wikipedia.org/wiki/UTF-16LE) or [UTF-32LE](https://en.wikipedia.org/wiki/UTF-32LE); compound character encoding schemes, such as [UTF-16](https://en.wikipedia.org/wiki/UTF-16), [UTF-32](https://en.wikipedia.org/wiki/UTF-32) and [ISO/IEC 2022](https://en.wikipedia.org/wiki/ISO/IEC_2022), switch between several simple schemes by using [byte order marks](https://en.wikipedia.org/wiki/Byte_order_mark) or [escape sequences](https://en.wikipedia.org/wiki/Escape_sequence); compressing schemes try to minimise the number of bytes used per code unit (such as [SCSU](https://en.wikipedia.org/wiki/Standard_Compression_Scheme_for_Unicode), [BOCU](https://en.wikipedia.org/wiki/Binary_Ordered_Compression_for_Unicode), and [Punycode](https://en.wikipedia.org/wiki/Punycode%22%20%5Co%20%22Punycode)).

Although [UTF-32BE](https://en.wikipedia.org/wiki/UTF-32BE) is a simpler CES, most systems working with Unicode use either [UTF-8](https://en.wikipedia.org/wiki/UTF-8), which is [backward compatible](https://en.wikipedia.org/wiki/Backward_compatibility) with fixed-width ASCII and maps Unicode code points to variable-width sequences of octets, or [UTF-16BE](https://en.wikipedia.org/wiki/UTF-16BE), which is [backward compatible](https://en.wikipedia.org/wiki/Backward_compatibility) with fixed-width UCS-2BE and maps Unicode code points to variable-width sequences of 16-bit words. See [comparison of Unicode encodings](https://en.wikipedia.org/wiki/Comparison_of_Unicode_encodings) for a detailed discussion.

Finally, there may be a **higher level protocol** which supplies additional information to select the particular variant of a [Unicode](https://en.wikipedia.org/wiki/Unicode) character, particularly where there are regional variants that have been 'unified' in Unicode as the same character. An example is the [XML](https://en.wikipedia.org/wiki/XML) attribute xml:lang.

The Unicode model uses the term **character map** for historical systems which directly assign a sequence of characters to a sequence of bytes, covering all of CCS, CEF and CES layers.[[5]](https://en.wikipedia.org/wiki/Character_encoding#cite_note-utr17-5)