3.2 Symbols

The Symbols area of the Unicode standard includes the encoding of symbolic characters, including punctuation, numbers, pictures for control codes, and dingbats.

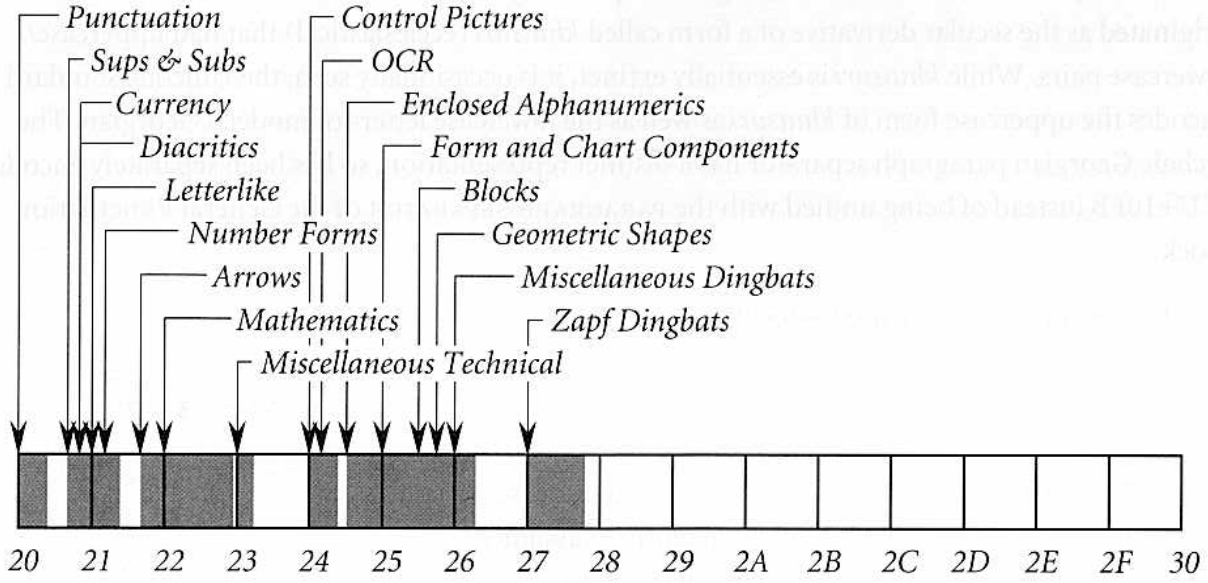


Figure 3-10. Symbols

General Punctuation U+2000 → U+206F

General Punctuation combines punctuation characters and character-like elements used to achieve certain text layout effects. Some punctuation characters can be used with many different scripts. Many general punctuation characters can also be found in the Unicode standard ASCII and Latin1 blocks.

In many cases, current standards include generic characters for punctuation instead of the more precisely specified characters used in printing. Examples include the single and double quotes, period, dash, and space. The Unicode standard includes these generic characters, but also encodes the unambiguous characters independently: various forms of quotation mark, decimal period, em-dash, en-dash, minus, hyphen, em-space, en-space, hair-space, zero-width space and so on.

Punctuation that is considered to belong to a specific script is found in the block corresponding to that script, such as U+03D7 GREEK QUESTION MARK ";" or the punctuation used with ideographs in the CJK Symbols block. Characters drawn with a dotted box are invisible in normal rendering.

Typographical Space Characters. Spaces all have the semantics of being word-break characters. Other than that, the main difference is in the width of the characters. U+2000 → U+2006 are standard quad widths used in typography. The figure space is provided for use in some languages as a thousands separator. The punctuation space is a space defined to be the same width as a period. The thin space and hair space are successively smaller-width spaces used for narrow word gaps and for justification of type. All of the fixed-width space characters are derived from conventional (hot lead) typography. Their functions are mostly replaced by algorithmic kerning and justification in computerized typography.

The *zero-width space* can be used in languages that have no visible word spacing in order to represent word-breaks, such as in Thai or Japanese. There are several varieties of zero-width spaces; the standard one is the *word-break space*, used to add soft word breaks in languages without word spaces. Additionally, there are two spaces which can be used in controlling cursive forms of characters, the *zero-width joiner* and *zero-width non-joiner*. There are also zero-width directional spaces, the *right-to-left zero-width space* and the *left-to-right zero-width space*. All of these properties are orthogonal: the *word-break space* does not affect joining or direction; the joiners neither cause a word-break nor have a direction; the directional spaces neither cause word-breaks nor affect joining.

Having a zero width makes the *zero-width space* similar in some respects to the zero-width layout characters; however, since it is used to delimit word breaks, it may be significant for searching or

sorting operations. See also U+0020 SPACE, U+00A0 NON-BREAKING SPACE, and U+3000 IDEOGRAPHIC SPACE.

Dashes. U+2010 HYPHEN is a unique character (unlike U+002D) that represents the hyphen as found in words such as "left-to-right." U+2011 NON-BREAKING HYPHEN and the U+2012 FIGURE DASH are present for compatibility with existing standards. The NON-BREAKING HYPHEN has the same semantic as U+2010 HYPHEN but should not be broken across lines. FIGURE DASH has the same (ambiguous) semantic as the U+002D HYPHEN-MINUS, but has the same width as digits (if they are monospaced). The EN DASH is used to indicate a range of values, such as 1973–1984. It should be distinguished from the U+2122 MINUS, which is an arithmetic operator. The U+2014 EM DASH is used to make a break—like this—in the flow of a sentence. It is commonly represented with a typewriter as a double-hyphen. In older mathematical typography, the EM DASH is also used to indicate a binary minus sign. A QUOTATION DASH is used to indicate the source of quotations. For general compatibility in interpreting formulas, the U+002D HYPHEN-MINUS, and FIGURE DASH should all be taken as indicating a minus sign, as in "x = a–b."

Quotation Marks. U+201A LOW SINGLE COMMA QUOTATION MARK, U+201E LOW DOUBLE COMMA QUOTATION MARK, U+2039 LEFT POINTING SINGLE GUILLEMET, and U+203A RIGHT POINTING SINGLE GUILLEMET have heterogeneous semantics. They may represent opening or closing quotation marks depending on which language they are used with.

Hyphenation Point. U+2027 HYPHENATION POINT is a raised dot used to indicate correct word breaking as in dic·tion·ar·ies. This is a punctuation mark, to be distinguished from U+00B7 MIDDLE DOT , which has multiple semantics.

Fraction Slash. U+2013 FRACTION SLASH is used between digits to represent rational values: 2/3, 3/9, and so on. Implementations may choose to change the size, shape and positioning of the digits and slash to reflect typographic concerns: such as representing 2/3 as a fraction similar in appearance to U+2154 FRACTION TWO THIRDS. When implementations choose to change the presentation of FRACTION SLASH and surrounding digits, NON-JOINER or a space (including thin spaces) can be used to separate digits that should not be included in the fraction.

Spacing Overscore. U+204E SPACING OVERSCORE corresponds to U+005F SPACING UNDERSCORE. It is a spacing character, not to be confused with U+0305 NON-SPACING OVERSCORE or U+0304 NON-SPACING MACRON. As with all over- or underscores, a sequence of these characters should connect in an unbroken line.

Zero-Width Layout Characters. In some circumstances, it is necessary for text formatting software to be able to specify whether or not adjacent characters may be grouped together. In the case of mixed left-to-right/right-to-left nested text runs, the formatting software must be able to specify the direction of characters that do not have an intrinsic directionality. For this purpose, the Unicode standard provides zero-width layout characters.

These characters are also significant in the Unicode bidirectional formatting algorithm (see Appendix A).

The Non-Joiner. U+200C ZERO WIDTH NON-JOINER is used to request that characters be rendered separately, when they would otherwise normally combine in some manner. For example, a ZERO WIDTH NON-JOINER between an "f" and an "i" will prevent an "fi" ligature from being displayed; a ZERO WIDTH NON-JOINER between an Arabic NOON and MEEM will prevent the normal cursive connection from being rendered, and a ZERO WIDTH NON-JOINER between an "a" and a NON-SPACING ACUTE will cause the NON-SPACING ACUTE to be displayed as a spacing character, and keep it from being superimposed on the "a" (that is, "a´" instead of "á"). The ZERO WIDTH NON-JOINER is also used in script-dependent ways; in Indic scripts, for example, to show the *virama* explicitly. (See the Devanagari block introduction.)

The Joiner. U+200D ZERO WIDTH JOINER is used to request that a character be rendered with a cursive connection when it otherwise would not. For example, to display the presentation form GLYPH FOR INITIAL ARABIC BAA (U+FE90), the Arabic letter *baa* can be followed by a ZERO WIDTH JOINER. The ZERO WIDTH JOINER does not have the semantic value of backspacing, and should not be used for overstriking characters. For example, *a-acute* is not correctly represented by *a-joiner-spacing acute.* The ZERO WIDTH JOINER can be used to indicate a tighter cursive connection between characters or to form a ligature (if available) when the default would be not to form one. On the other hand, the ZERO WIDTH JOINER can be placed between already cursively-connected text with no effect: thus Arabic *baa-joiner-meem* will have the same appearance as *baa meem.* The ZERO WIDTH JOINER also has other uses in some scripts, such as Tibetan. (See the Tibetan block description.)

Left-to-Right and Right-to-Left Marks. U+200E LEFT-TO-RIGHT MARK and U+200F RIGHT-TO-LEFT MARK are treated by directional layout algorithms as though they were normal left-to-right or right-to-left characters, but can be used to achieve many special effects because they are invisible in rendering. (See Appendix A on bidirectional character encoding.)

Layout Characters. Except for their effect on the layout of the text in which they are contained, the zero-width layout characters can be treated just as any other character by the processing software; in particular they do not introduce a mode or state into the character sequence. For any non-layout text processing, such as sorting, searching, and so on, the zero-width layout characters can simply be filtered out.

Bidirectional Ordering Codes. These codes are used in the Bidirectional Formatting Algorithm, described in Appendix A. Systems that handle bidirectional scripts (Arabic and Hebrew) should be sensitive to these codes. The codes include:

U+202A	Left-to-Right Embedding	(LRE)
U+202B	Right-to-Left Embedding	(RLE)
U+202C	Pop Directional Formatting	(PDF)
U+202D	Left-to-Right Override	(RLO)
U+202E	Right-to-Left Override	(LRO)

As with the other zero-width character codes, except for their effect on the layout of the text in which they are contained, the bidirectional ordering characters can be treated just as any other character by the processing software. For non-layout text processing, such as sorting, searching and so on, the zero-width layout characters can simply be filtered out. However, when modifying text, care should be taken to maintain these correctly, since the matching pairs of zero-width formatting characters must be coördinated. (See Appendix A.)

Line and Paragraph Separator. For historical reasons, carriage-return and line-feed are not used consistantly across different systems. The Unicode standard provides (and encourages use of) the *line* and *paragraph separator* characters to provide clear information about where line and paragraph boundaries occur. A paragraph separator indicates where a new paragraph should start. This could cause, for example, the line to be broken, the inter-paragraph line spacing to be applied, and indentation of the first line. A line separator indicates that a line-break should occur at this point; although the text continues on the next line, it does not start a new paragraph: no inter-paragraph line spacing nor paragraphic indentation is applied. Since these are separator codes, it is not necessary to start the first line or paragraph, or end the last line or paragraph with them.

Encoding Structure. The Unicode block for General Punctuation is divided into the following ranges:

U+2000 → U+200A	Typographical space characters
U+200B	Zero-width space
U+200C → U+200F	Zero-width layout characters
U+2010 → U+2027	Printing punctuation characters
U+2028 → U+2029	Line and paragraph separators
U+202A → U+202E	Bidirectional ordering codes
U+202F	Currently unassigned
U+2030 → U+2044	Printing punctuation characters
U+2045 → U+206F	Currently unassigned

Superscripts and Subscripts U+2070 → U+209F

Superscripts and subscripts have been included in the Unicode standard solely to provide compatibility with existing character sets. In general, the Unicode character encoding does not attempt to describe the positioning of a character above or below the baseline in typographical layout. The superscript digits one, two, and three are coded in the Latin1 block.

Standards. The characters in this block are from sets registered with ECMA under ISO 2375 for use with ISO 2022.

Encoding Structure. The Unicode block for Superscripts and Subscripts is divided into the following ranges:

U+2070	Superscript zero
U+2071 → U+2073	Currently unassigned
U+2074 → U+2079	Superscript numbers
U+207A → U+207F	Superscript mathematical operators
U+2080 → U+2089	Subscript numbers
U+208A → U+208E	Subscript mathematical operators
U+208F → U+209F	Currently unassigned

Currency Symbols U+20A0 → U+20CF

This block contains currency symbols not encoded in other blocks. Where the Unicode standard follows the layout of an existing standard, such as for the ASCII, Latin1 and Thai blocks, the currency symbols are encoded in those blocks, rather than here.

Unification. The Unicode standard does not duplicate encodings where more than one currency is expressed with the same symbol. Many currency symbols are overstruck letters. There are therefore many minor variants, such as the U+0024 DOLLAR SIGN $, $\mathbb{S}$, or $\S$, with one or two vertical bars, or other graphical variation. The Unicode standard considers these variants to be typographical and provides a single encoding.

Claims that glyph variants of a certain currency symbol are used consistently to indicate a particular currency could not be substantiated upon further research. Please refer to ISO DIS 10367, Annex B (informative) for an example of multiple renderings for U+00A3 POUND SIGN.

Encoding Structure. The Unicode block for Currency Symbols is divided into the following ranges:

U+20A0 → U+20AA	Currency symbols
U+20AB → U+20CF	Currently unassigned

The following table lists common currency symbols encoded in other blocks.

Dollar, milreis, escudo	U+0024	DOLLAR SIGN
Cent	U+00A2	CENT SIGN
Pound	U+00A3	POUND SIGN
General currency	U+00A4	CURRENCY SIGN
Yen or yuan	U+00A5	YEN SIGN
Dutch florin	U+0192	LATIN SMALL LETTER SCRIPT F
Baht	U+0E3F	THAI BAHT SIGN

Diacritical Marks for Symbols U+20D0 → U+20FF

Diacritical marks for symbols are generally applied to mathematical or technical symbols. These can be used to extend the range of the symbol set. For example, U+20D3 NON-SPACING SHORT VERTICAL BAR can be used to express negation. Its presentation may change in those circumstances, changing length or slant. That is, U+2261 IDENTICAL TO, followed by U+20D3 is equivalent to U+2262 NOT IDENTICAL TO. In this case, there was a precomposed form for the negated symbol. However, this is not always true, and U+20D3 can be used with other symbols to form the negation. For example, U+2258 CORRESPONDS TO followed by U+20D3 can be used to express *does not correspond to*, without requiring that a precomposed form be part of the Unicode standard.

Other non-spacing characters can also be used in mathematical expressions, of course. For example, a U+0304 NON-SPACING MACRON is commonly used in propostional logic to indicate logical negation.

Enclosing Diacritics. These non-spacing characters are supplied for compatibility with existing standards, allowing individual base characters to be enclosed in several ways. For example, U+2460 CIRCLED DIGIT ONE ① can be expressed as U+0030 DIGIT ONE "1" + U+20DD ENCLOSING CIRCLE ⃝. As with other non-spacing characters, this one can also be applied productively; *circled letter alef* can be produced by the sequence: U+05D0 HEBREW LETTER ALEPH א + U+20DD ENCLOSING CIRCLE ⃝. The non-spacing enclosing diacritics cannot be used to enclose a sequence of base characters. For example, there is no way to represent U+246A CIRCLED NUMBER ELEVEN with the ENCLOSING CIRCLE, since there is no single character NUMBER ELEVEN.

Encoding Structure. The Unicode block for Diacritical Marks for Symbols is divided into the following ranges:

U+20D0 → U+20E1	Symbol diacritics
U+20E2 → U+20FF	Currently unassigned

Letterlike Symbols U+2100 → U+214F

Letterlike symbols are symbols which are derived in some way from ordinary letters of an alphabetic script. The Unicode standard includes symbols here based on Latin, Greek, and Hebrew letters. They are distinct from ordinary letters in that they do not have the alphabetic character property and do not normally collate in alphabetic sequence. They may also have different directional properties from normal letters; for example, the four transfinite cardinal symbols (U+2135 → U+2138) are used in ordinary mathematical text and do not share the strong right-to-left directionality of the Hebrew letters they are derived from.

Styles. The letterlike symbols constitute one of the few instances in which the Unicode standard encodes stylistic variants of letters as distinct characters. For example, there are instances of black letter, double-struck, and script styles for certain Latin letters used as mathematical symbols. The choice of these stylistic variants for encoding reflects their common use as distinct symbols. It is recognized that a particular style can be applied to any Latin letter with a resulting semantic distinction in mathematical or logical text; applications which require such systematic stylistic semantics should achieve them by using styles directly, rather than by seeking to extend the character-by-character encoding of such variants in the Unicode standard.

The black-letter style is often referred to as *Fraktur* or *Gothic* in various sources. Technically, Fraktur and Gothic typefaces are distinct designs from black letter, but no encoding distinctions are implied in the various symbol sources. The Unicode standard simply uses black letter forms as the archetypes.

A similar consideration applies to the double-struck style. This style is not literally double-struck, but is instead an open outline design which gives the visual appearance of being struck twice with a horizontal shift. For encoding purposes this style can be considered equivalent to letterlike symbols rendered in outlined or shadowed typefaces to carry conventional semantic distinctions.

The Unicode standard does not encode serif versus sans-serif styles distinctly among letterlike symbols. This style is always of typographical significance only, and never carries a semantic distinction.

Standards. The Unicode standard encodes letterlike symbols from many different national standards and corporate collections.

Encoding Structure. The Unicode block for Letterlike Symbols is divided into the following ranges:

U+2100 → U+2138	Letterlike symbols
U+2139 → U+214F	Currently unassigned

Number Forms U+2150 → U+218F

Number Form characters are presented solely for compatibility with existing standards. The fractions can be equivalently represented with the U+2044 FRACTION SLASH. The Roman numerals can be composed of sequences of the appropriate Latin letters. U+2180 ROMAN NUMERAL ONE THOUSAND C D and U+216F ROMAN NUMERAL ONE THOUSAND are actually variants of the same glyph, but are distinguished because of existing standards; similarly, the upper- and lowercase variants of Roman numerals have been separately encoded. U+2181 ROMAN NUMERAL FIVE THOUSAND, and U+2182 ROMAN NUMERAL TEN THOUSAND are useful characters, since they represent characters used in Roman numerals that do not have good substitutes elsewhere in the Unicode standard.

Encoding Structure. The Unicode block for Number Forms is divided into the following ranges:

U+2150 → U+2152	Currently unassigned
U+2153 → U+215F	Vulgar fractions
U+2160 → U+2182	Roman numerals and small roman numerals
U+2183 → U+218F	Currently unassigned

Arrows U+2190 → U+21FF

Arrows are used for a variety of purposes: to imply directional relation, logical derivation or implication, or to represent the cursor control keys.

The Unicode standard attempts to provide fairly complete encodings for generic arrow shapes, especially where there are established usages with well defined semantics; the Unicode standard does not attempt to encode separately every possible stylistic variant of arrows, especially where their use is mainly decorative. For most arrow variants, the Unicode standard provides encodings in the two horizontal directions, often in the four cardinal directions. For the single and double arrows the Unicode standard provides encodings in eight directions.

Standards. The Unicode standard encodes arrows from many different national standards and corporate collections.

Unifications. Arrows expressing mathematical relations have been encoded in the arrows block. For example, U+21D2 RIGHT DOUBLE ARROW ⇒ may be the equivalent of *implies.*

Long and short arrow forms encoded in glyph standards or typesetting systems such as T_EX are not represented by separate Unicode values.

Encoding Principles. Because the arrows have such a wide variety of applications, there may be several semantic values for the same Unicode character value: for example, U+21B5 DOWNWARD ARROW WITH CORNER LEFT ↵ may be the equivalent of *carriage return*; U+2191 UP ARROW ↑ may be the equivalent of *increases* or *exponent.*

Encoding Structure. The Unicode block for arrows is divided into the following ranges:

U+2190 → U+21EA	Arrows
U+21EB → U+21FF	Currently unassigned

Mathematical Operators U+2200 → U+22FF

The Mathematical Operators block includes character encodings for operators, relations, geometric symbols, and a few other symbols with special usages confined largely to mathematical contexts.

In addition to the characters in this block, mathematical operators are also found in the ASCII and Latin1 blocks. A few of the symbols from the Miscellaneous Technical block, and characters from General Punctuation are also used in mathematical notation.

Latin letters in special font styles and used as mathematical operators, such as U+2118 SCRIPT P ℘, as well as the Hebrew letter *alef* used as the operator U+2135 FIRST TRANSFINITE CARDINAL ℵ, are encoded in the block for letterlike symbols.

Standards. Many national standards' mathematical operators are covered by the characters encoded in this block. These standards include such special collections as ANSI Y10.20, ISO DIS 6862.2, ISO 8879, and the collection of the American Mathematical Society, as well as the original repertoire of TEX.

Encoding Principles. Mathematical operators often have more than one meaning. Therefore the encoding of this block is intentionally shape-based, with numerous instances in which several semantic values can be attributed to the same Unicode value. For example, U+2218 ∘ RING OPERATOR may be the equivalent of *white small circle* or *composite function* or *apl jot*. The Unicode standard does not attempt to distinguish all the possible semantic values which may be applied to these symbols.

On the other hand, mathematical operators, and especially relation symbols, may appear in various standards, handbooks, and fonts with a large number of purely graphical variants. Where variants were recognizable as such from the sources, they were not encoded separately.

Unifications. Mathematical operators such as U+21D2 IMPLIES ⇒ and U+2194 IF AND ONLY IF ↔ have been unified with the corresponding arrows in the Arrows block.

The operator U+2208 ELEMENT OF is occasionally rendered with a taller shape than shown here. Mathematical handbooks and standards consulted treat these as variants of the same glyph. U+220A SMALL ELEMENT OF is separately encoded, because some existing standards distinguish it from U+2208.

The operators U+226B MUCH GREATER THAN and U+226A MUCH LESS THAN are sometimes rendered in a nested shape. Since no semantic distinction applies, the Unicode standard provides a single encoding for each of these operators.

A large class of unifications applies to variants of relation symbols involving equality, similarity, and/or negation. Variants involving one- or two-barred *equal signs*, one- or two-*tilde similarity signs*, and vertical or slanted *negation slashes* and *negation slashes* of different lengths are not separately encoded. Thus, for example, U+2288 NEITHER A SUBSET OF NOR EQUAL TO, is the prototype for at least six different glyph variants noted in various collections.

There are two instances in which essentially stylistic variants are separately encoded: U+2265 GREATER THAN OR EQUAL TO ≥ is distinguished from U+2267 GREATER THAN OVER EQUAL TO ≧; the same distinction applies to LESS THAN OR EQUAL TO. This exception to the general rule regarding variation results from character mapping requirements to some Asian standards which distinguish the two forms.

Greek-Derived Symbols. Several mathematical operators derived from Greek characters have been given separate encodings to match usage in existing standards. These operators may occasionally occur in context with variables using the same characters, or are used typographically quite distinct from normal Greek letters. These operators include U+2206 INCREMENT Δ, U+220F N-ARY PRODUCT ∏, and U+2211 N-ARY SUMMATION ∑.

Other duplicated Greek characters are those for U+00B5 MICRO SIGN μ in the Latin1 block and U+2126 OHM Ω in Letterlike symbols. All other Greek characters with special mathematical semantics have been unified with the Greek characters in the Greek block since their mathematical semantics do not distinguish them substantially from Greek letters.

Miscellaneous Symbols. U+2212 MINUS SIGN − is a mathematical operator, to be distinguished from the ASCII-derived U+002D HYPHEN-MINUS -, which may look the same as minus sign, or may be shorter in length. U+22EE → U+22F1 are a set of ellipses used in matrix notation.

Encoding structure. The Unicode block for Mathematical Operators is divided into the following ranges:

U+2200 → U+22F1	Mathematical operators
U+22F2 → U+22FF	Currently unassigned

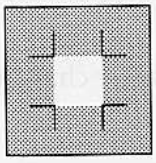
Miscellaneous Technical U+2300 → U+23FF

This block encodes technical symbols including keytop labels such as U+232B DELETE TO THE LEFT KEY. Excluded from consideration were symbols that are not normally used in one-dimensional text, but are intended for two-dimensional diagrammatic use, such as symbols for electronic circuits. An unusually large expansion space is provided since it is anticipated that there are a large number of technical symbols that were not considered in the first version of the Unicode standard.

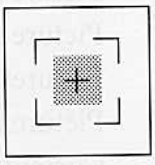
Encoding Structure. The Unicode block for Miscellaneous Technical symbols is divided into the following ranges:

U+2300 → U+2301	APL symbols
U+2302 → U+2307	Miscellaneous symbols
U+2308 → U+230B	Ceilings and floors
U+230C → U+230F	Crops
U+2310 → U+2317	Miscellaneous symbols
U+2318	Keyboard symbol
U+2319 → U+231B	Miscellaneous symbols
U+231C → U+231F	Quine corners
U+2320 → U+2321	Partial math symbols for compatibility
U+2322 → U+2323	Miscellaneous symbols
U+2324 → U+2328	Keyboard symbols
U+2329 → U+232A	Bra and Ket
U+232B	Keyboard symbol
U+232C	Benzene ring
U+232D → U+23FF	Currently unassigned

The usage of crops and quine corners is as indicated in this diagram:



Use of crops



Use of quine corners

Pictures for Control Codes U+2400 → U+243F

The need to show the presence of the C0 control codes and the SPACE unequivocally when data is displayed has led to conventional representations for these non-graphic characters.

By definition, control codes themselves are manifested only by their action. However, it is sometimes necessary to show the position of a control code within a data stream. Conventional illustrations for the ASCII C0 control codes have been developed.

By definition, the SPACE is a blank graphic. Conventions have also been established for the explicit representation of the SPACE.

Standards. The CNS 11643 standard encodes characters for pictures of control codes. Standard representations for control characters have been defined, for example, in ANSI X3.32 and ISO 2047, but for the control code graphics U+2400 → U+241F only the semantic is encoded in the Unicode standard. This allows a particular application to use the graphic representation it prefers.

Pictures for ASCII Space. Two specific glyphs are provided that may be used to represent the ASCII space character (U+2420 and U+2422).

Code Points for Pictures for Control Codes. The remaining code points in this block are not associated with specific glyphs, but rather are available to encode *any* desired pictorial representation of the given control code. The assumption is that the particular pictures used to represent control codes are often specific to different systems, and are not often the subject of text interchange between systems.

Encoding Structure. The Unicode block for Pictures for Control Codes is divided into the following ranges:

U+2400 → U+241F	Code points for pictures for control codes U+0000 → U+001F
U+2420	Picture for the ASCII space character
U+2421	Picture for *delete*
U+2422 → U+2423	Pictures for the ASCII space character
U+2424	Picture for *new line*
U+2425 → U+243F	Currently unassigned

Optical Character Recognition U+2440 → U+245F

This block includes the symbolic characters of the OCR-A character set that do not correspond to ASCII characters, and magnetic ink character recognition (MICR) symbols used in check processing.

Standards. Both sets of symbols are specified in ISO 2033.

Encoding Structure. The Unicode block for Optical Character Recognitions is divided into the following ranges:

U+2440 → U+2445	OCR-A symbols
U+2446 → U+244A	MICR symbols
U+244B → U+245F	Currently unassigned

Enclosed Alphanumerics U+2460 → U+24FF

The enclosed numbers and Latin letters of this block come from several sources, chiefly East Asian standards, and are provided for compatibility with them.

Standards. Enclosed letters and numbers occur in the Korean National Standard, KS C 5601, and in the Chinese national standard, GB 2312, as well as in various East Asian industry standards.

The Zapf Dingbats character set contains four sets of encircled numbers (including encircled zero). The black on white set that has numbers with serifs is encoded here (U+2460 → U+2468, and U+24EA). The other three sets are encoded in the range U+2776 → U+2793 in the Zapf Dingbats block.

Decompositions. The parenthesized letters or numbers may be decomposed to a sequence of opening parenthesis, letter or digit(s), closing parenthesis. The numbers with period may be decomposed to digit(s), followed by a period. The encircled letters and single digit numbers may be decomposed to letter or digit followed by U+20DD ENCLOSING CIRCLE. The encircled numbers 10 though 20 may not be decomposed.

Encoding Structure. The Unicode block for Enclosed Alphanumerics is divided into the following ranges:

U+2460 → U+2473	Encircled numbers 1–20
U+2474 → U+2487	Parenthesized numbers 1–20
U+2488 → U+249B	Numbers with period 1–20
U+249C → U+24B5	Parenthesized small Latin a–z
U+24B6 → U+24CF	Encircled capital Latin A–Z
U+24D0 → U+24E9	Encircled small Latin a–z
U+24EA	Encircled number 0
U+24EB → U+24FF	Currently unassigned

Form and Chart Components U+2500 → U+257F

The characters in the Form and Chart Components block are encoded solely for compatibility with existing standards.

Standards. GB 2312, KS C 5601 and industry standards.

Encoding structure. The Unicode block for Form and Chart Components is divided into the following ranges:

U+2500 → U+254F	Single line box and line drawing elements
U+2550 → U+256C	Line box drawing elements with double line segments
U+256D → U+2570	Curved corner segments
U+2571 → U+2573	Diagonal line segments and miscellaneous
U+2574 → U+257F	Line end pieces and connectors

Blocks U+2580 → U+259F

The Blocks represent a graphic compatibility zone in the Unicode standard. A number of existing national and vendor standards, including IBM PC Code Page 437, contain a number of characters intended to enable a simple kind of character cell graphic by filling some fraction of each cell, or by filling each character cell by some degree of shading. The Unicode standard does not encourage this kind of character-based graphics model, but includes a minimal set of such characters encoded for backwards compatibility with the existing standards.

Half-block fill characters are included for each half of a character cell, plus a graduated series of vertical and horizontal fractional fills based on one-eighth parts. Also included are a series of shades based on one-quarter shadings. The fractional fills do not form a logically complete set, but are only intended for backwards compatibility; future versions of the Unicode standard should not extend this set.

Encoding Structure. The Unicode block for Blocks is divided into the following ranges:

U+2580 → U+2590	Character cell fractional fill characters
U+2591 → U+2593	Percent shade characters
U+2594 → U+2595	More character cell fractional fill characters
U+2596 → U+259F	Currently unassigned

Geometric Shapes U+25A0 → U+25FF

The Geometric Shapes are a collection of characters intended to encode prototypes for various commonly used geometrical shapes—mostly squares, triangles, and circles. The collection is somewhat arbitrary in scope; it is a compendium of shapes from various character and glyph standards. The typical distinctions more systematically encoded include black versus white, large versus small, basic shape (square versus triangle versus circle), orientation, and top versus bottom or left versus right part.

The hatched and cross-hatched squares at U+25A4 → U+25A9 derive from the Korean national standard (KS C 5601), in which they were probably intended as representations of fill patterns; however, since the semantics of those characters are insufficiently defined in that standard, the Unicode character encoding simply carries the glyphs themselves as geometric shapes to provide a mapping for that standard.

U+25CA LOZENGE ◊ is a typographical symbol seen in PostScript and in the Macintosh character set. It should be distinguished both from the generic U+25C7 WHITE DIAMOND and the U+2662 WHITE DIAMOND SUIT, as well as from another character sometimes called a lozenge: U+2311 SQUARE LOZENGE.

The squares and triangles at U+25E7 → U+25EE are derived from the Linotype font collection.

Standards. The Geometric Shapes are derived from a large range of national and vendor character standards.

Encoding Structure. The Unicode block for Geometric Shapes is divided into the following ranges:

U+25A0 → U+25AB	Geometric shapes based on squares
U+25AC → U+25AF	Geometric shapes based on rectangles
U+25B0 → U+25B1	Geometric shapes based on parallelograms
U+25B2 → U+25C5	Geometric shapes based on triangles
U+25C6 → U+25C8	Geometric shapes based on diamonds
U+25C9 → U+25CA	Geometric shapes (miscellaneous)
U+25CB → U+25E1	Geometric shapes based on circles and arcs
U+25E2 → U+25E5	Geometric shapes based on right triangles
U+25E6	Geometric shape based on bullet
U+25E7 → U+25EB	Geometric shapes based on squares
U+25EC → U+25EE	Geometric shapes based on triangles
U+25EF → U+25FF	Currently unassigned

Miscellaneous Dingbats U+2600 → U+26FF

The Miscellaneous Dingbats block consists of a very heterogenous collection of symbols which do not fit in any other Unicode block, and which tend to be rather pictographic in nature. The term "dingbat" is borrowed from the Zapf Dingbats (see the block starting at U+2700), and has come to mean any of a large number of non-alphabetic picture-like symbols which fall outside the more conventional sets of mathematical and technical symbols. The usage of dingbats is typically text-decorative, but they may also be seen treated as normal text characters in such textual applications as typesetting of chess books, card game manuals, horoscopes and so on.

Characters in the Miscellaneous Dingbats set can be rendered in more than one way, unlike characters in the Zapf Dingbats block, in which characters correspond to an explicit glyph. U+2641 EARTH, and U+2645 URANUS, which belong to the Miscellaneous Dingbats set, both have alternative glyphs.

The order of the Miscellaneous Dingbats is completely arbitrary, but an attempt has been made to keep like symbols together and to group subsets of them into meaningful orders. Some of these subsets include weather and astronomical symbols, pointing hands, religious and ideological symbols, the I Ching trigrams; planet and zodiacal symbols, chess pieces, card suits, and musical dingbats. For other moon phases, see Geometric Shapes.

Corporate logos and collections of pictures of animals, vehicles, foods, and so on are not included since they tend either to be very specific in usage (logos, political party symbols) or nonconventional in appearance and semantic interpretation (pictures of cows or cats; fizzing champagne bottles), and hence are inappropriate for encoding as characters. The Unicode standard recommends that such items be incorporated in text via higher protocols which allow intermixing of graphic images with text, rather than by indefinite extension of the number of Miscellaneous Dingbats encoded as characters. However, a large unassigned space has been set aside in the Miscellaneous Dingbats block with the expectation that other conventional sets of such symbols will be found appropriate for character encoding in the future.

Note in particular that the musical dingbats are just that—dingbats—a small set of text decorative characters. No attempt is made to provide a complete character encoding for musical notation. The Unicode standard considers musical notation to be a higher-order text format which requires two-dimensional layout control and complex structures.

Standards. The Miscellaneous Dingbats are derived from a large range of national and vendor character standards.

Encoding Structure. The Unicode block for Miscellaneous Dingbats is divided into the following ranges:

U+2600 → U+2603	Weather symbols
U+2604 → U+2613	Miscellaneous dingbats
U+2614 → U+2619	Currently unassigned
U+261A → U+262F	Miscellaneous dingbats
U+2630 → U+2637	I-Ching symbols
U+2638 → U+263C	Miscellaneous dingbats
U+263D → U+2644	Moon phases and planets
U+2645 → U+2647	Miscellaneous dingbats
U+2648 → U+2653	Signs of the zodiac
U+2654 → U+265F	Chess pieces
U+2660 → U+2667	Card suits
U+2668	Miscellaneous dingbat
U+2669 → U+266F	Musical symbols
U+2670 → U+26FF	Currently unassigned

Zapf Dingbats U+2700 → U+27BF

The Zapf Dingbats are a well-established set of symbols comprising the industry standard "Zapf Dingbat" font—currently available in most laser printers. Other series of Zapf Dingbats also exist, but are not encoded in the Unicode standard because they are not widely implemented in existing hardware and software as character-encoded fonts. Dingbats that are part of other standards have been encoded in the context of Geometrical Forms and Shapes, Encircled Alphanumerics, and Miscellaneous Dingbats. The order of the remaining dingbats follows the PostScript encoding.

The Zapf Dingbats differ in their treatment in the Unicode standard from all other characters. They are encoded as absolutely specific glyph shapes, rather than as glyphic archetypes for abstract characters which can be represented in different faces and styles. Thus it would be incorrect to arbitrarily replace U+279D TRIANGLE-HEADED RIGHT ARROW → with any other right arrow dingbat or with any of the generic arrows from the Unicode Arrows block (U+2190 → U+21FF). In other words, since the Zapf Dingbats refer to glyphs from a specific typeface, their semantic value *is* their shape.

Unifications. A number of the Zapf Dingbats represent shapes which overlap with regular Unicode symbol characters. Instead of coding both a Zapf Dingbat glyph shape and a separate character whose glyphic representation is normally indistinguishable from that shape, the Unicode standard unifies the two. The characters in question include: card suits, BLACK STAR, BLACK TELEPHONE, and BLACK RIGHT-POINTING INDEX (see Miscellaneous Dingbats); BLACK CIRCLE and BLACK SQUARE (see Geometric Shapes); white encircled numbers 1 to 10 (see Enclosed Alphanumerics); and several generic arrows (see Arrows).

The positions of these unified characters are left unassigned in the Zapf Dingbats block and are cross-referenced to the assigned positions in the other blocks. Applications are free to choose alternative glyphs for representing those characters (as for any normal Unicode characters), including, of course, the exact shapes required for rendering them in the Zapf Dingbat font on an imaging device.

To illustrate this distinction, an application encoding an encircled digit one with U+2460 ① CIRCLED DIGIT ONE may allow for the rendition of that encircled digit in any appropriate typeface — serif or sans serif, roman or italic, and with the circle rendered in different thicknesses. On the other hand, an application encoding an encircled digit one with the Zapf Dingbat U+2780 SANS-SERIF CIRCLED DIGIT ONE ➀ requires an explicit sans serif glyph from the Zapf Dingbat font for rendering.

Encoding Structure. The Unicode block for Zapf Dingbats is divided into the following ranges:

U+2700	Currently unassigned
U+2701 → U+27BE	ITC Zapf Dingbats, series 100
U+27BF	Currently unassigned