

!!!!!!! DO NOT EDIT THIS FILE !!!!!!! # This file is machine-generated by lib/unicore/mktables from the Unicode # database, Version 5.2.0. Any changes made here will be lost!

To change this file, edit lib/unicore/mktables instead.

NAME

perluniprops - Index of Unicode Version 5.2.0 properties in Perl

DESCRIPTION

There are many properties in Unicode, and Perl provides access to almost all of them, as well as some additional extensions and short-cut synonyms.

And just about all of the few that aren't accessible through the Perl core are accessible through the modules: `Unicode::Normalize` and `Unicode::UCD`, and for Unihan properties, via the CPAN module `Unicode::Unihan`.

This document merely lists all available properties and does not attempt to explain what each property really means. There is a brief description of each Perl extension. There is some detail about `Blocks`, `Scripts`, `General_Category`, and `Bidi_Class` in *perlunicode*, but to find out about the intricacies of the Unicode properties, refer to the Unicode standard. A good starting place is <http://www.unicode.org/reports/tr44/>. More information on the Perl extensions is in *perlrecharclass*.

Note that you can define your own properties; see *"User-Defined Character Properties"* in *perlunicode*.

Properties accessible through `\p{}` and `\P{}`

The Perl regular expression `\p{}` and `\P{}` constructs give access to most of the Unicode character properties. The table below shows all these constructs, both single and compound forms.

Compound forms consist of two components, separated by an equals sign or a colon. The first component is the property name, and the second component is the particular value of the property to match against, for example, `\p{Script: Greek}` or `\p{Script=Greek}` both mean to match characters whose `Script` property is `Greek`.

Single forms, like `\p{Greek}`, are mostly Perl-defined shortcuts for their equivalent compound forms. The table shows these equivalences. (In our example, `\p{Greek}` is a just a shortcut for `\p{Script=Greek}`.) There are also a few Perl-defined single forms that are not shortcuts for a compound form. One such is `\p{Word}`. These are also listed in the table.

In parsing these constructs, Perl always ignores Upper/lower case differences everywhere within the `{braces}`. Thus `\p{Greek}` means the same thing as `\p{greek}`. But note that changing the case of the 'p' or 'P' before the left brace completely changes the meaning of the construct, from "match" (for `\p{}`) to "doesn't match" (for `\P{}`). Casing in this document is for improved legibility.

Also, white space, hyphens, and underscores are also normally ignored everywhere between the `{braces}`, and hence can be freely added or removed even if the `/x` modifier hasn't been specified on the regular expression. But a 'T' at the beginning of an entry in the table below means that tighter (stricter) rules are used for that entry:

Single form (`\p{name}`) tighter rules:

White space, hyphens, and underscores ARE significant except for:

- * white space adjacent to a non-word character
- * underscores separating digits in numbers

That means, for example, that you can freely add or remove white space adjacent to (but within) the braces without affecting the meaning.

Compound form (`\p{name=value}` or `\p{name:value}`) tighter rules:

The tighter rules given above for the single form apply to everything to the right of the colon or equals; the looser rules still apply to everything to the left.

That means, for example, that you can freely add or remove white space adjacent to (but within) the braces and the colon or equal sign.

Some properties are considered obsolete, but still available. There are several varieties of obsolescence:

Obsolete

Properties marked with an 'O' in the table are considered obsolete. At the time of this writing (Unicode version 5.2) there is no information in the Unicode standard about the implications of a property being obsolete.

Stabilized

Obsolete properties may be stabilized. This means that they are not actively maintained by Unicode, and will not be extended as new characters are added to the standard. Such properties are marked with an 'S' in the table. At the time of this writing (Unicode version 5.2) there is no further information in the Unicode standard about the implications of a property being stabilized.

Deprecated

Obsolete properties may be deprecated. This means that their use is strongly discouraged, so much so that a warning will be issued if used, unless the regular expression is in the scope of a `no warnings 'deprecated'` statement. A 'D' flags each such entry in the table, and the entry there for the longest, most descriptive version of the property will give the reason it is deprecated, and perhaps advice. Perl may issue such a warning, even for properties that aren't officially deprecated by Unicode, when there used to be characters or code points that were matched by them, but no longer. This is to warn you that your program may not work like it did on earlier Unicode releases.

A deprecated property may be made unavailable in a future Perl version, so it is best to move away from them.

Some Perl extensions are present for backwards compatibility and are discouraged from being used, but not obsolete. An 'X' flags each such entry in the table.

Matches in the Block property have shortcuts that begin with 'In_'. For example, `\p{Block=Latin1}` can be written as `\p{In_Latin1}`. For backward compatibility, if there is no conflict with another shortcut, these may also be written as `\p{Latin1}` or `\p{Is_Latin1}`. But, N.B., there are numerous such conflicting shortcuts. Use of these forms for Block is discouraged, and are flagged as such, not only because of the potential confusion as to what is meant, but also because a later release of Unicode may preempt the shortcut, and your program would no longer be correct. Use the 'In_' form instead to avoid this, or even more clearly, use the compound form, e.g., `\p{blk:latin1}`. See *"Blocks" in perlunicode* for more information about this.

The table below has two columns. The left column contains the `\p{}` constructs to look up, possibly preceeded by the flags mentioned above; and the right column contains information about them, like a description, or synonyms. It shows both the single and compound forms for each property that has them. If the left column is a short name for a property, the right column will give its longer, more descriptive name; and if the left column is the longest name, the right column will show any equivalent shortest name, in both single and compound forms if applicable.

The right column will also caution you if a property means something different than what might normally be expected.

All single forms are Perl extensions; a few compound forms are as well, and are noted as such.

Numbers in (parentheses) indicate the total number of code points matched by the property. For emphasis, those properties that match no code points at all are listed as well in a separate section

following the table.

There is no description given for most non-Perl defined properties (See <http://www.unicode.org/reports/tr44/> for that).

For compactness, '*' is used as a wildcard instead of showing all possible combinations. For example, entries like:

```
\p{Gc: *} \p{General_Category: *}
```

mean that 'Gc' is a synonym for 'General_Category', and anything that is valid for the latter is also valid for the former. Similarly,

```
\p{Is_*} \p{*}
```

means that if and only if, for example, `\p{Foo}` exists, then `\p{Is_Foo}` and `\p{IsFoo}` are also valid and all mean the same thing. And similarly, `\p{Foo=Bar}` means the same as `\p{Is_Foo=Bar}` and `\p{IsFoo=Bar}`. '*' here is restricted to something not beginning with an underscore.

Also, in binary properties, 'Yes', 'T', and 'True' are all synonyms for 'Y'. And 'No', 'F', and 'False' are all synonyms for 'N'. The table shows 'Y*' and 'N*' to indicate this, and doesn't have separate entries for the other possibilities. Note that not all properties which have values 'Yes' and 'No' are binary, and they have all their values spelled out without using this wild card, and a NOT clause in their description that highlights their not being binary. These also require the compound form to match them, whereas true binary properties have both single and compound forms available.

Note that all non-essential underscores are removed in the display of the short names below.

Summary legend:

* is a wild-card

(ld+) in the info column gives the number of code points matched by this property.

D means this is deprecated.

O means this is obsolete.

S means this is stabilized.

T means tighter (stricter) name matching applies.

X means use of this form is discouraged.

NAME	INFO
X <code>\p{Aegean_Numbers}</code>	<code>\p{Block=Aegean_Numbers}</code> (64)
T <code>\p{Age: 1.1}</code>	Code point's usage introduced in version 1.1 (33_979)
T <code>\p{Age: 2.0}</code>	Code point's usage was introduced in version 2.0; See also Property 'Present_In' (144_521)
T <code>\p{Age: 2.1}</code>	Code point's usage was introduced in version 2.1; See also Property 'Present_In' (2)
T <code>\p{Age: 3.0}</code>	Code point's usage was introduced in version 3.0; See also Property 'Present_In' (10_307)
T <code>\p{Age: 3.1}</code>	Code point's usage was introduced in version 3.1; See also Property 'Present_In' (44_978)
T <code>\p{Age: 3.2}</code>	Code point's usage was introduced in

	version 3.2; See also Property 'Present_In' (1016)
T \p{Age: 4.0}	Code point's usage was introduced in version 4.0; See also Property 'Present_In' (1226)
T \p{Age: 4.1}	Code point's usage was introduced in version 4.1; See also Property 'Present_In' (1273)
T \p{Age: 5.0}	Code point's usage was introduced in version 5.0; See also Property 'Present_In' (1369)
T \p{Age: 5.1}	Code point's usage was introduced in version 5.1; See also Property 'Present_In' (1624)
T \p{Age: 5.2}	Code point's usage was introduced in version 5.2; See also Property 'Present_In' (6648)
\p{Age: Unassigned}	Code point's usage has not been assigned in any Unicode release thus far. (867_169)
\p{AHex}	\p{ASCII_Hex_Digit} (= \p{ASCII_Hex_Digit= Y}) (22)
\p{AHex: *}	\p{ASCII_Hex_Digit: *}
\p{All}	\p{Any} (1_114_112)
\p{Alnum}	Alphabetic and (Decimal) Numeric (100_931)
\p{Alpha}	\p{Alphabetic=Y} (100_520)
\p{Alpha: *}	\p{Alphabetic: *}
\p{Alphabetic}	\p{Alpha} (= \p{Alphabetic=Y}) (100_520)
\p{Alphabetic: N*}	(Short: \p{Alpha=N}, \p{Alpha}) (1_013_592)
\p{Alphabetic: Y*}	(Short: \p{Alpha=Y}, \p{Alpha}) (100_520)
X \p{Alphabetic_Presentation_Forms}	\p{Block= Alphabetic_Presentation_Forms} (80)
X \p{Ancient_Greek_Musical_Notation}	\p{Block= Ancient_Greek_Musical_Notation} (80)
X \p{Ancient_Greek_Numbers}	\p{Block=Ancient_Greek_Numbers} (80)
X \p{Ancient_Symbols}	\p{Block=Ancient_Symbols} (64)
\p{Any}	[\x{0000}-\x{10FFFF}] (1_114_112)
\p{Arab}	\p{Arabic} (= \p{Script=Arabic}) (NOT \p{Block=Arabic}) (1030)
\p{Arabic}	\p{Script=Arabic} (Short: \p{Arab}; NOT \p{Block=Arabic}) (1030)
X \p{Arabic_Presentation_Forms_A}	\p{Block= Arabic_Presentation_Forms_A} (688)
X \p{Arabic_Presentation_Forms_B}	\p{Block= Arabic_Presentation_Forms_B} (144)
X \p{Arabic_Supplement}	\p{Block=Arabic_Supplement} (48)
\p{Armenian}	\p{Script=Armenian} (Short: \p{Armn}; NOT \p{Block=Armenian}) (90)
\p{Armi}	\p{Imperial_Aramaic} (= \p{Script= Imperial_Aramaic}) (NOT \p{Block= Imperial_Aramaic}) (31)
\p{Armn}	\p{Armenian} (= \p{Script=Armenian}) (NOT \p{Block=Armenian}) (90)
X \p{Arrows}	\p{Block=Arrows} (112)
\p{ASCII}	\p{Block=Basic_Latin} [[:ASCII:]] (128)
\p{ASCII_Hex_Digit}	\p{ASCII_Hex_Digit=Y} (Short: \p{AHex})

	(22)
<code>\p{ASCII_Hex_Digit: N*}</code>	(Short: <code>\p{AHex=N}</code> , <code>\P{AHex}</code>) (1_114_090)
<code>\p{ASCII_Hex_Digit: Y*}</code>	(Short: <code>\p{AHex=Y}</code> , <code>\P{AHex}</code>) (22)
<code>\p{Assigned}</code>	All assigned code points (246_877)
<code>\p{Avestan}</code>	<code>\p{Script=Avestan}</code> (Short: <code>\p{Avst}</code> ; NOT <code>\p{Block=Avestan}</code>) (61)
<code>\p{Avst}</code>	<code>\p{Avestan}</code> (= <code>\p{Script=Avestan}</code>) (NOT <code>\p{Block=Avestan}</code>) (61)
<code>\p{Bali}</code>	<code>\p{Balinese}</code> (= <code>\p{Script=Balinese}</code>) (NOT <code>\p{Block=Balinese}</code>) (121)
<code>\p{Balinese}</code>	<code>\p{Script=Balinese}</code> (Short: <code>\p{Bali}</code> ; NOT <code>\p{Block=Balinese}</code>) (121)
<code>\p{Bamu}</code>	<code>\p{Bamum}</code> (= <code>\p{Script=Bamum}</code>) (NOT <code>\p{Block=Bamum}</code>) (88)
<code>\p{Bamum}</code>	<code>\p{Script=Bamum}</code> (Short: <code>\p{Bamu}</code> ; NOT <code>\p{Block=Bamum}</code>) (88)
X <code>\p{Basic_Latin}</code>	<code>\p{ASCII}</code> (= <code>\p{Block=Basic_Latin}</code>) (128)
<code>\p{Bc: *}</code>	<code>\p{Bidi_Class: *}</code>
<code>\p{Beng}</code>	<code>\p{Bengali}</code> (= <code>\p{Script=Bengali}</code>) (NOT <code>\p{Block=Bengali}</code>) (92)
<code>\p{Bengali}</code>	<code>\p{Script=Bengali}</code> (Short: <code>\p{Beng}</code> ; NOT <code>\p{Block=Bengali}</code>) (92)
<code>\p{Bidi_C}</code>	<code>\p{Bidi_Control}</code> (= <code>\p{Bidi_Control=Y}</code>) (7)
<code>\p{Bidi_C: *}</code>	<code>\p{Bidi_Control: *}</code>
<code>\p{Bidi_Class: AL}</code>	<code>\p{Bidi_Class=Arabic_Letter}</code> (1116)
<code>\p{Bidi_Class: AN}</code>	<code>\p{Bidi_Class=Arabic_Number}</code> (48)
<code>\p{Bidi_Class: Arabic_Letter}</code>	(Short: <code>\p{Bc=AL}</code>) (1116)
<code>\p{Bidi_Class: Arabic_Number}</code>	(Short: <code>\p{Bc=AN}</code>) (48)
<code>\p{Bidi_Class: B}</code>	<code>\p{Bidi_Class=Paragraph_Separator}</code> (7)
<code>\p{Bidi_Class: BN}</code>	<code>\p{Bidi_Class=Boundary_Neutral}</code> (4016)
<code>\p{Bidi_Class: Boundary_Neutral}</code>	(Short: <code>\p{Bc=BN}</code>) (4016)
<code>\p{Bidi_Class: Common_Separator}</code>	(Short: <code>\p{Bc=CS}</code>) (15)
<code>\p{Bidi_Class: CS}</code>	<code>\p{Bidi_Class=Common_Separator}</code> (15)
<code>\p{Bidi_Class: EN}</code>	<code>\p{Bidi_Class=European_Number}</code> (131)
<code>\p{Bidi_Class: ES}</code>	<code>\p{Bidi_Class=European_Separator}</code> (12)
<code>\p{Bidi_Class: ET}</code>	<code>\p{Bidi_Class=European_Terminator}</code> (63)
<code>\p{Bidi_Class: European_Number}</code>	(Short: <code>\p{Bc=EN}</code>) (131)
<code>\p{Bidi_Class: European_Separator}</code>	(Short: <code>\p{Bc=ES}</code>) (12)
<code>\p{Bidi_Class: European_Terminator}</code>	(Short: <code>\p{Bc=ET}</code>) (63)
<code>\p{Bidi_Class: L}</code>	<code>\p{Bidi_Class=Left_To_Right}</code> (1_099_541)
<code>\p{Bidi_Class: Left_To_Right}</code>	(Short: <code>\p{Bc=L}</code>) (1_099_541)
<code>\p{Bidi_Class: Left_To_Right_Embedding}</code>	(Short: <code>\p{Bc=LRE}</code>) (1)
<code>\p{Bidi_Class: Left_To_Right_Override}</code>	(Short: <code>\p{Bc=LRO}</code>) (1)
<code>\p{Bidi_Class: LRE}</code>	<code>\p{Bidi_Class=Left_To_Right_Embedding}</code> (1)
<code>\p{Bidi_Class: LRO}</code>	<code>\p{Bidi_Class=Left_To_Right_Override}</code> (1)
<code>\p{Bidi_Class: Nonspacing_Mark}</code>	(Short: <code>\p{Bc=NSM}</code>) (1173)
<code>\p{Bidi_Class: NSM}</code>	<code>\p{Bidi_Class=Nonspacing_Mark}</code> (1173)
<code>\p{Bidi_Class: ON}</code>	<code>\p{Bidi_Class=Other_Neutral}</code> (3523)
<code>\p{Bidi_Class: Other_Neutral}</code>	(Short: <code>\p{Bc=ON}</code>) (3523)
<code>\p{Bidi_Class: Paragraph_Separator}</code>	(Short: <code>\p{Bc=B}</code>) (7)
<code>\p{Bidi_Class: PDF}</code>	<code>\p{Bidi_Class=Pop_Directional_Format}</code> (1)
<code>\p{Bidi_Class: Pop_Directional_Format}</code>	(Short: <code>\p{Bc=PDF}</code>) (1)
<code>\p{Bidi_Class: R}</code>	<code>\p{Bidi_Class=Right_To_Left}</code> (4441)
<code>\p{Bidi_Class: Right_To_Left}</code>	(Short: <code>\p{Bc=R}</code>) (4441)
<code>\p{Bidi_Class: Right_To_Left_Embedding}</code>	(Short: <code>\p{Bc=RLE}</code>) (1)
<code>\p{Bidi_Class: Right_To_Left_Override}</code>	(Short: <code>\p{Bc=RLO}</code>) (1)

```

\p{Bidi_Class: RLE}      \p{Bidi_Class=Right_To_Left_Embedding} (1)
\p{Bidi_Class: RLO}      \p{Bidi_Class=Right_To_Left_Override} (1)
\p{Bidi_Class: S}        \p{Bidi_Class=Segment_Separator} (3)
\p{Bidi_Class: Segment_Separator} (Short: \p{Bc=S}) (3)
\p{Bidi_Class: White_Space} (Short: \p{Bc=WS}) (18)
\p{Bidi_Class: WS}       \p{Bidi_Class=White_Space} (18)
\p{Bidi_Control}         \p{Bidi_Control=Y} (Short: \p{BidiC}) (7)
\p{Bidi_Control: N*}     (Short: \p{BidiC=N}, \p{BidiC}) (1_114_105)
\p{Bidi_Control: Y*}     (Short: \p{BidiC=Y}, \p{BidiC}) (7)
\p{Bidi_M}               \p{Bidi_Mirrored} (= \p{Bidi_Mirrored=Y})
                          (543)
\p{Bidi_M: *}            \p{Bidi_Mirrored: *}
\p{Bidi_Mirrored}        \p{Bidi_Mirrored=Y} (Short: \p{BidiM})
                          (543)
\p{Bidi_Mirrored: N*}    (Short: \p{BidiM=N}, \p{BidiM}) (1_113_569)
\p{Bidi_Mirrored: Y*}    (Short: \p{BidiM=Y}, \p{BidiM}) (543)
\p{Blank}                \h, Horizontal white space (19)
\p{Blk: *}               \p{Block: *}
\p{Block: Aegean_Numbers} (Single: \p{InAegeanNumbers}) (64)
\p{Block: Alphabetic_Presentation_Forms} (Single:
                          \p{InAlphabeticPresentationForms}) (80)
\p{Block: Ancient_Greek_Musical_Notation} (Single:
                          \p{InAncientGreekMusicalNotation}) (80)
\p{Block: Ancient_Greek_Numbers} (Single:
                          \p{InAncientGreekNumbers}) (80)
\p{Block: Ancient_Symbols} (Single: \p{InAncientSymbols}) (64)
\p{Block: Arabic}         (Single: \p{InArabic}; NOT \p{Arabic} NOR
                          \p{Is_Arabic}) (256)
\p{Block: Arabic_Presentation_Forms_A} (Single:
                          \p{InArabicPresentationFormsA}) (688)
\p{Block: Arabic_Presentation_Forms_B} (Single:
                          \p{InArabicPresentationFormsB}) (144)
\p{Block: Arabic_Supplement} (Single: \p{InArabicSupplement}) (48)
\p{Block: Armenian}      (Single: \p{InArmenian}; NOT \p{Armenian}
                          NOR \p{Is_Armenian}) (96)
\p{Block: Arrows}        (Single: \p{InArrows}) (112)
\p{Block: ASCII}         \p{Block=Basic_Latin} (128)
\p{Block: Avestan}       (Single: \p{InAvestan}; NOT \p{Avestan}
                          NOR \p{Is_Avestan}) (64)
\p{Block: Balinese}      (Single: \p{InBalinese}; NOT \p{Balinese}
                          NOR \p{Is_Balinese}) (128)
\p{Block: Bamum}         (Single: \p{InBamum}; NOT \p{Bamum} NOR
                          \p{Is_Bamum}) (96)
\p{Block: Basic_Latin}   (Short: \p{Blk=ASCII}, \p{ASCII}) (128)
\p{Block: Bengali}       (Single: \p{InBengali}; NOT \p{Bengali}
                          NOR \p{Is_Bengali}) (128)
\p{Block: Block_Elements} (Single: \p{InBlockElements}) (32)
\p{Block: Bopomofo}     (Single: \p{InBopomofo}; NOT \p{Bopomofo}
                          NOR \p{Is_Bopomofo}) (48)
\p{Block: Bopomofo_Extended} (Single: \p{InBopomofoExtended}) (32)
\p{Block: Box_Drawing}   (Single: \p{InBoxDrawing}) (128)
\p{Block: Braille_Patterns} (Single: \p{InBraillePatterns}) (256)
\p{Block: Buginese}     (Single: \p{InBuginese}; NOT \p{Buginese}
                          NOR \p{Is_Buginese}) (32)
\p{Block: Buhid}        (Single: \p{InBuhid}; NOT \p{Buhid} NOR
                          \p{Is_Buhid}) (32)
    
```

`\p{Block: Byzantine_Musical_Symbols}` (Single: `\p{InByzantineMusicalSymbols}`) (256)
`\p{Block: Canadian_Syllabics}` `\p{Block=`
 `Unified_Canadian_Aboriginal_Syllabics}`
 (640)
`\p{Block: Carian}` (Single: `\p{InCarian}`; NOT `\p{Carian}` NOR
 `\p{Is_Carian}`) (64)
`\p{Block: Cham}` (Single: `\p{InCham}`; NOT `\p{Cham}` NOR
 `\p{Is_Cham}`) (96)
`\p{Block: Cherokee}` (Single: `\p{InCherokee}`; NOT `\p{Cherokee}`
 NOR `\p{Is_Cherokee}`) (96)
`\p{Block: CJK_Compatibility}` (Single: `\p{InCJKCompatibility}`) (256)
`\p{Block: CJK_Compatibility_Forms}` (Single:
 `\p{InCJKCompatibilityForms}`) (32)
`\p{Block: CJK_Compatibility_Ideographs}` (Single:
 `\p{InCJKCompatibilityIdeographs}`) (512)
`\p{Block: CJK_Compatibility_Ideographs_Supplement}` (Single:
 `\p{InCJKCompatibilityIdeographs-`
 `Supplement}`) (544)
`\p{Block: CJK_Radicals_Supplement}` (Single:
 `\p{InCJKRadicalsSupplement}`) (128)
`\p{Block: CJK_Strokes}` (Single: `\p{InCJKStrokes}`) (48)
`\p{Block: CJK_Symbols_And_Punctuation}` (Single:
 `\p{InCJKSymbolsAndPunctuation}`) (64)
`\p{Block: CJK_Unified_Ideographs}` (Single:
 `\p{InCJKUnifiedIdeographs}`) (20_992)
`\p{Block: CJK_Unified_Ideographs_Extension_A}` (Single:
 `\p{InCJKUnifiedIdeographsExtensionA}`)
 (6592)
`\p{Block: CJK_Unified_Ideographs_Extension_B}` (Single:
 `\p{InCJKUnifiedIdeographsExtensionB}`)
 (42_720)
`\p{Block: CJK_Unified_Ideographs_Extension_C}` (Single:
 `\p{InCJKUnifiedIdeographsExtensionC}`)
 (4160)
`\p{Block: Combining_Diacritical_Marks}` (Single:
 `\p{InCombiningDiacriticalMarks}`) (112)
`\p{Block: Combining_Diacritical_Marks_For_Symbols}` (Short: `\p{Blk=`
 `CombiningMarksForSymbols}`,
 `\p{InCombiningMarksForSymbols}`) (48)
`\p{Block: Combining_Diacritical_Marks_Supplement}` (Single:
 `\p{InCombiningDiacriticalMarks-`
 `Supplement}`) (64)
`\p{Block: Combining_Half_Marks}` (Single: `\p{InCombiningHalfMarks}`)
 (16)
`\p{Block: Combining_Marks_For_Symbols}` `\p{Block=`
 `Combining_Diacritical_Marks_For_Symbols}`
 (48)
`\p{Block: Common_Indic_Number_Forms}` (Single:
 `\p{InCommonIndicNumberForms}`) (16)
`\p{Block: Control_Pictures}` (Single: `\p{InControlPictures}`) (64)
`\p{Block: Coptic}` (Single: `\p{InCoptic}`; NOT `\p{Coptic}` NOR
 `\p{Is_Coptic}`) (128)
`\p{Block: Counting_Rod_Numerals}` (Single:
 `\p{InCountingRodNumerals}`) (32)
`\p{Block: Cuneiform}` (Single: `\p{InCuneiform}`; NOT

```

        \p{Cuneiform} NOR \p{Is_Cuneiform})
        (1024)
\p{Block: Cuneiform_Numbers_And_Punctuation} (Single:
        \p{InCuneiformNumbersAndPunctuation})
        (128)
\p{Block: Currency_Symbols} (Single: \p{InCurrencySymbols}) (48)
\p{Block: Cypriot_Syllabary} (Single: \p{InCypriotSyllabary}) (64)
\p{Block: Cyrillic} (Single: \p{InCyrillic}; NOT \p{Cyrillic}
        NOR \p{Is_Cyrillic}) (256)
\p{Block: Cyrillic_Extended_A} (Single: \p{InCyrillicExtendedA})
        (32)
\p{Block: Cyrillic_Extended_B} (Single: \p{InCyrillicExtendedB})
        (96)
\p{Block: Cyrillic_Supplement} (Single: \p{InCyrillicSupplement})
        (48)
\p{Block: Cyrillic_Supplementary} \p{Block=Cyrillic_Supplement}
        (48)
\p{Block: Deseret} (Single: \p{InDeseret}) (80)
\p{Block: Devanagari} (Single: \p{InDevanagari}; NOT
        \p{Devanagari} NOR \p{Is_Devanagari})
        (128)
\p{Block: Devanagari_Extended} (Single: \p{InDevanagariExtended})
        (32)
\p{Block: Dingbats} (Single: \p{InDingbats}) (192)
\p{Block: Domino_Tiles} (Single: \p{InDominoTiles}) (112)
\p{Block: Egyptian_Hieroglyphs} (Single:
        \p{InEgyptianHieroglyphs}; NOT
        \p{Egyptian_Hieroglyphs} NOR
        \p{Is_Egyptian_Hieroglyphs}) (1072)
\p{Block: Enclosed_Alphanumeric_Supplement} (Single:
        \p{InEnclosedAlphanumericSupplement})
        (256)
\p{Block: Enclosed_Alphanumerics} (Single:
        \p{InEnclosedAlphanumerics}) (160)
\p{Block: Enclosed_CJK_Letters_And_Months} (Single:
        \p{InEnclosedCJKLettersAndMonths}) (256)
\p{Block: Enclosed_Ideographic_Supplement} (Single:
        \p{InEnclosedIdeographicSupplement})
        (256)
\p{Block: Ethiopic} (Single: \p{InEthiopic}; NOT \p{Ethiopic}
        NOR \p{Is_Ethiopic}) (384)
\p{Block: Ethiopic_Extended} (Single: \p{InEthiopicExtended}) (96)
\p{Block: Ethiopic_Supplement} (Single: \p{InEthiopicSupplement})
        (32)
\p{Block: General_Punctuation} (Single: \p{InGeneralPunctuation})
        (112)
\p{Block: Geometric_Shapes} (Single: \p{InGeometricShapes}) (96)
\p{Block: Georgian} (Single: \p{InGeorgian}; NOT \p{Georgian}
        NOR \p{Is_Georgian}) (96)
\p{Block: Georgian_Supplement} (Single: \p{InGeorgianSupplement})
        (48)
\p{Block: Glagolitic} (Single: \p{InGlagolitic}; NOT
        \p{Glagolitic} NOR \p{Is_Glagolitic})
        (96)
\p{Block: Gothic} (Single: \p{InGothic}; NOT \p{Gothic} NOR
        \p{Is_Gothic}) (32)
    
```



```

\p{Block: Greek}          \p{Block=Greek_And_Coptic} (NOT \p{Greek}
                          NOR \p{Is_Greek}) (144)
\p{Block: Greek_And_Coptic} (Short: \p{Blk=Greek}, \p{InGreek};
                             NOT \p{Greek} NOR \p{Is_Greek}) (144)
\p{Block: Greek_Extended} (Single: \p{InGreekExtended}) (256)
\p{Block: Gujarati}      (Single: \p{InGujarati}; NOT \p{Gujarati}
                          NOR \p{Is_Gujarati}) (128)
\p{Block: Gurmukhi}      (Single: \p{InGurmukhi}; NOT \p{Gurmukhi}
                          NOR \p{Is_Gurmukhi}) (128)
\p{Block: Halfwidth_And_Fullwidth_Forms} (Single:
                                          \p{InHalfwidthAndFullwidthForms}) (240)
\p{Block: Hangul_Compatibility_Jamo} (Single:
                                       \p{InHangulCompatibilityJamo}) (96)
\p{Block: Hangul_Jamo}   (Single: \p{InHangulJamo}) (256)
\p{Block: Hangul_Jamo_Extended_A} (Single:
                                    \p{InHangulJamoExtendedA}) (32)
\p{Block: Hangul_Jamo_Extended_B} (Single:
                                    \p{InHangulJamoExtendedB}) (80)
\p{Block: Hangul_Syllables} (Single: \p{InHangulSyllables})
                             (11_184)
\p{Block: Hanunoo}      (Single: \p{InHanunoo}; NOT \p{Hanunoo}
                          NOR \p{Is_Hanunoo}) (32)
\p{Block: Hebrew}      (Single: \p{InHebrew}; NOT \p{Hebrew} NOR
                          \p{Is_Hebrew}) (112)
\p{Block: High_Private_Use_Surrogates} (Single:
                                          \p{InHighPrivateUseSurrogates}) (128)
\p{Block: High_Surrogates} (Single: \p{InHighSurrogates}) (896)
\p{Block: Hiragana}     (Single: \p{InHiragana}; NOT \p{Hiragana}
                          NOR \p{Is_Hiragana}) (96)
\p{Block: Ideographic_Description_Characters} (Single:
                                                \p{InIdeographicDescriptionCharacters})
                                                (16)
\p{Block: Imperial_Aramaic} (Single: \p{InImperialAramaic}; NOT
                              \p{Imperial_Aramaic} NOR
                              \p{Is_Imperial_Aramaic}) (32)
\p{Block: Inscriptional_Pahlavi} (Single:
                                   \p{InInscriptionalPahlavi}; NOT
                                   \p{Inscriptional_Pahlavi} NOR
                                   \p{Is_Inscriptional_Pahlavi}) (32)
\p{Block: Inscriptional_Parthian} (Single:
                                    \p{InInscriptionalParthian}; NOT
                                    \p{Inscriptional_Parthian} NOR
                                    \p{Is_Inscriptional_Parthian}) (32)
\p{Block: IPA_Extensions} (Single: \p{InIPAExtensions}) (96)
\p{Block: Javanese}      (Single: \p{InJavanese}; NOT \p{Javanese}
                          NOR \p{Is_Javanese}) (96)
\p{Block: Kaithi}        (Single: \p{InKaithi}; NOT \p{Kaithi} NOR
                          \p{Is_Kaithi}) (80)
\p{Block: Kanbun}        (Single: \p{InKanbun}) (16)
\p{Block: Kangxi_Radicals} (Single: \p{InKangxiRadicals}) (224)
\p{Block: Kannada}       (Single: \p{InKannada}; NOT \p{Kannada}
                          NOR \p{Is_Kannada}) (128)
\p{Block: Katakana}      (Single: \p{InKatakana}; NOT \p{Katakana}
                          NOR \p{Is_Katakana}) (96)
\p{Block: Katakana_Phonetic_Extensions} (Single:
                                          \p{InKatakanaPhoneticExtensions}) (16)

```

`\p{Block: Kayah_Li}` (Single: `\p{InKayahLi}`) (48)
`\p{Block: Kharoshthi}` (Single: `\p{InKharoshthi}`; NOT
`\p{Kharoshthi}` NOR `\p{Is_Kharoshthi}`)
(96)
`\p{Block: Khmer}` (Single: `\p{InKhmer}`; NOT `\p{Khmer}` NOR
`\p{Is_Khmer}`) (128)
`\p{Block: Khmer_Symbols}` (Single: `\p{InKhmerSymbols}`) (32)
`\p{Block: Lao}` (Single: `\p{InLao}`; NOT `\p{Lao}` NOR
`\p{Is_Lao}`) (128)
`\p{Block: Latin_1}` `\p{Block=Latin_1_Supplement}` (128)
`\p{Block: Latin_1_Supplement}` (Short: `\p{Blk=Latin1}`,
`\p{InLatin1}`) (128)
`\p{Block: Latin_Extended_A}` (Single: `\p{InLatinExtendedA}`) (128)
`\p{Block: Latin_Extended_Additional}` (Single:
`\p{InLatinExtendedAdditional}`) (256)
`\p{Block: Latin_Extended_B}` (Single: `\p{InLatinExtendedB}`) (208)
`\p{Block: Latin_Extended_C}` (Single: `\p{InLatinExtendedC}`) (32)
`\p{Block: Latin_Extended_D}` (Single: `\p{InLatinExtendedD}`) (224)
`\p{Block: Lepcha}` (Single: `\p{InLepcha}`; NOT `\p{Lepcha}` NOR
`\p{Is_Lepcha}`) (80)
`\p{Block: Letterlike_Symbols}` (Single: `\p{InLetterlikeSymbols}`)
(80)
`\p{Block: Limbu}` (Single: `\p{InLimbu}`; NOT `\p{Limbu}` NOR
`\p{Is_Limbu}`) (80)
`\p{Block: Linear_B_Ideograms}` (Single: `\p{InLinearBIdeograms}`)
(128)
`\p{Block: Linear_B_Syllabary}` (Single: `\p{InLinearBSyllabary}`)
(128)
`\p{Block: Lisu}` (Single: `\p{InLisu}`) (48)
`\p{Block: Low_Surrogates}` (Single: `\p{InLowSurrogates}`) (1024)
`\p{Block: Lycian}` (Single: `\p{InLycian}`; NOT `\p{Lycian}` NOR
`\p{Is_Lycian}`) (32)
`\p{Block: Lydian}` (Single: `\p{InLydian}`; NOT `\p{Lydian}` NOR
`\p{Is_Lydian}`) (32)
`\p{Block: Mahjong_Tiles}` (Single: `\p{InMahjongTiles}`) (48)
`\p{Block: Malayalam}` (Single: `\p{InMalayalam}`; NOT
`\p{Malayalam}` NOR `\p{Is_Malayalam}`) (128)
`\p{Block: Mathematical_Alphanumeric_Symbols}` (Single:
`\p{InMathematicalAlphanumericSymbols}`)
(1024)
`\p{Block: Mathematical_Operators}` (Single:
`\p{InMathematicalOperators}`) (256)
`\p{Block: Meetei_Mayek}` (Single: `\p{InMeeteiMayek}`; NOT
`\p{Meetei_Mayek}` NOR
`\p{Is_Meetei_Mayek}`) (64)
`\p{Block: Miscellaneous_Mathematical_Symbols_A}` (Single:
`\p{InMiscellaneousMathematicalSymbolsA}`)
(48)
`\p{Block: Miscellaneous_Mathematical_Symbols_B}` (Single:
`\p{InMiscellaneousMathematicalSymbolsB}`)
(128)
`\p{Block: Miscellaneous_Symbols}` (Single:
`\p{InMiscellaneousSymbols}`) (256)
`\p{Block: Miscellaneous_Symbols_And_Arrows}` (Single:
`\p{InMiscellaneousSymbolsAndArrows}`)
(256)

```

\p{Block: Miscellaneous_Technical} (Single:
    \p{InMiscellaneousTechnical}) (256)
\p{Block: Modifier_Tone_Letters} (Single:
    \p{InModifierToneLetters}) (32)
\p{Block: Mongolian} (Single: \p{InMongolian}; NOT
    \p{Mongolian} NOR \p{Is_Mongolian}) (176)
\p{Block: Musical_Symbols} (Single: \p{InMusicalSymbols}) (256)
\p{Block: Myanmar} (Single: \p{InMyanmar}; NOT \p{Myanmar}
    NOR \p{Is_Myanmar}) (160)
\p{Block: Myanmar_Extended_A} (Single: \p{InMyanmarExtendedA}) (32)
\p{Block: New_Tai_Lue} (Single: \p{InNewTaiLue}; NOT
    \p{New_Tai_Lue} NOR \p{Is_New_Tai_Lue})
    (96)
\p{Block: NKo} (Single: \p{InNko}; NOT \p{Nko} NOR
    \p{Is_Nko}) (64)
\p{Block: No_Block} (Single: \p{InNoBlock}) (864_192)
\p{Block: Number_Forms} (Single: \p{InNumberForms}) (64)
\p{Block: Ogham} (Single: \p{InOgham}; NOT \p{Ogham} NOR
    \p{Is_Ogham}) (32)
\p{Block: Ol_Chiki} (Single: \p{InOlChiki}) (48)
\p{Block: Old_Italic} (Single: \p{InOldItalic}; NOT
    \p{Old_Italic} NOR \p{Is_Old_Italic})
    (48)
\p{Block: Old_Persian} (Single: \p{InOldPersian}; NOT
    \p{Old_Persian} NOR \p{Is_Old_Persian})
    (64)
\p{Block: Old_South_Arabian} (Single: \p{InOldSouthArabian}) (32)
\p{Block: Old_Turkic} (Single: \p{InOldTurkic}; NOT
    \p{Old_Turkic} NOR \p{Is_Old_Turkic})
    (80)
\p{Block: Optical_Character_Recognition} (Single:
    \p{InOpticalCharacterRecognition}) (32)
\p{Block: Oriya} (Single: \p{InOriya}; NOT \p{Oriya} NOR
    \p{Is_Oriya}) (128)
\p{Block: Osmanya} (Single: \p{InOsmanya}; NOT \p{Osmanya}
    NOR \p{Is_Osmanya}) (48)
\p{Block: Phags_Pa} (Single: \p{InPhagsPa}; NOT \p{Phags_Pa}
    NOR \p{Is_Phags_Pa}) (64)
\p{Block: Phaistos_Disc} (Single: \p{InPhaistosDisc}) (48)
\p{Block: Phoenician} (Single: \p{InPhoenician}; NOT
    \p{Phoenician} NOR \p{Is_Phoenician})
    (32)
\p{Block: Phonetic_Extensions} (Single: \p{InPhoneticExtensions})
    (128)
\p{Block: Phonetic_Extensions_Supplement} (Single:
    \p{InPhoneticExtensionsSupplement}) (64)
\p{Block: Private_Use} \p{Block=Private_Use_Area} (NOT
    \p{Private_Use} NOR \p{Is_Private_Use})
    (6400)
\p{Block: Private_Use_Area} (Short: \p{Blk=PrivateUse},
    \p{InPrivateUse}; NOT \p{Private_Use}
    NOR \p{Is_Private_Use}) (6400)
\p{Block: Rejang} (Single: \p{InRejang}; NOT \p{Rejang} NOR
    \p{Is_Rejang}) (48)
\p{Block: Rumi_Numeral_Symbols} (Single: \p{InRumiNumeralSymbols})
    (32)

```

<code>\p{Block: Runic}</code>	(Single: <code>\p{InRunic}</code> ; NOT <code>\p{Runic}</code> NOR <code>\p{Is_Runic}</code>) (96)
<code>\p{Block: Samaritan}</code>	(Single: <code>\p{InSamaritan}</code> ; NOT <code>\p{Samaritan}</code> NOR <code>\p{Is_Samaritan}</code>) (64)
<code>\p{Block: Saurashtra}</code>	(Single: <code>\p{InSaurashtra}</code> ; NOT <code>\p{Saurashtra}</code> NOR <code>\p{Is_Saurashtra}</code>) (96)
<code>\p{Block: Shavian}</code>	(Single: <code>\p{InShavian}</code>) (48)
<code>\p{Block: Sinhala}</code>	(Single: <code>\p{InSinhala}</code> ; NOT <code>\p{Sinhala}</code> NOR <code>\p{Is_Sinhala}</code>) (128)
<code>\p{Block: Small_Form_Variants}</code>	(Single: <code>\p{InSmallFormVariants}</code>) (32)
<code>\p{Block: Spacing_Modifier_Letters}</code>	(Single: <code>\p{InSpacingModifierLetters}</code>) (80)
<code>\p{Block: Specials}</code>	(Single: <code>\p{InSpecials}</code>) (16)
<code>\p{Block: Sundanese}</code>	(Single: <code>\p{InSundanese}</code> ; NOT <code>\p{Sundanese}</code> NOR <code>\p{Is_Sundanese}</code>) (64)
<code>\p{Block: Superscripts_And_Subscripts}</code>	(Single: <code>\p{InSuperscriptsAndSubscripts}</code>) (48)
<code>\p{Block: Supplemental_Arrows_A}</code>	(Single: <code>\p{InSupplementalArrowsA}</code>) (16)
<code>\p{Block: Supplemental_Arrows_B}</code>	(Single: <code>\p{InSupplementalArrowsB}</code>) (128)
<code>\p{Block: Supplemental_Mathematical_Operators}</code>	(Single: <code>\p{InSupplementalMathematicalOperators}</code>) (256)
<code>\p{Block: Supplemental_Punctuation}</code>	(Single: <code>\p{InSupplementalPunctuation}</code>) (128)
<code>\p{Block: Supplementary_Private_Use_Area_A}</code>	(Single: <code>\p{InSupplementaryPrivateUseAreaA}</code>) (65_536)
<code>\p{Block: Supplementary_Private_Use_Area_B}</code>	(Single: <code>\p{InSupplementaryPrivateUseAreaB}</code>) (65_536)
<code>\p{Block: Syloti_Nagri}</code>	(Single: <code>\p{InSylotiNagri}</code> ; NOT <code>\p{Syloti_Nagri}</code> NOR <code>\p{Is_Syloti_Nagri}</code>) (48)
<code>\p{Block: Syriac}</code>	(Single: <code>\p{InSyriac}</code> ; NOT <code>\p{Syriac}</code> NOR <code>\p{Is_Syriac}</code>) (80)
<code>\p{Block: Tagalog}</code>	(Single: <code>\p{InTagalog}</code> ; NOT <code>\p{Tagalog}</code> NOR <code>\p{Is_Tagalog}</code>) (32)
<code>\p{Block: Tagbanwa}</code>	(Single: <code>\p{InTagbanwa}</code> ; NOT <code>\p{Tagbanwa}</code> NOR <code>\p{Is_Tagbanwa}</code>) (32)
<code>\p{Block: Tags}</code>	(Single: <code>\p{InTags}</code>) (128)
<code>\p{Block: Tai_Le}</code>	(Single: <code>\p{InTaiLe}</code> ; NOT <code>\p{Tai_Le}</code> NOR <code>\p{Is_Tai_Le}</code>) (48)
<code>\p{Block: Tai_Tham}</code>	(Single: <code>\p{InTaiTham}</code> ; NOT <code>\p{Tai_Tham}</code> NOR <code>\p{Is_Tai_Tham}</code>) (144)
<code>\p{Block: Tai_Viet}</code>	(Single: <code>\p{InTaiViet}</code> ; NOT <code>\p{Tai_Viet}</code> NOR <code>\p{Is_Tai_Viet}</code>) (96)
<code>\p{Block: Tai_Xuan_Jing_Symbols}</code>	(Single: <code>\p{InTaiXuanJingSymbols}</code>) (96)
<code>\p{Block: Tamil}</code>	(Single: <code>\p{InTamil}</code> ; NOT <code>\p{Tamil}</code> NOR <code>\p{Is_Tamil}</code>) (128)
<code>\p{Block: Telugu}</code>	(Single: <code>\p{InTelugu}</code> ; NOT <code>\p{Telugu}</code> NOR <code>\p{Is_Telugu}</code>) (128)

<code>\p{Block: Thaana}</code>	<code>(Single: \p{InThaana}; NOT \p{Thaana} NOR \p{Is_Thaana}) (64)</code>
<code>\p{Block: Thai}</code>	<code>(Single: \p{InThai}; NOT \p{Thai} NOR \p{Is_Thai}) (128)</code>
<code>\p{Block: Tibetan}</code>	<code>(Single: \p{InTibetan}; NOT \p{Tibetan} NOR \p{Is_Tibetan}) (256)</code>
<code>\p{Block: Tifinagh}</code>	<code>(Single: \p{InTifinagh}; NOT \p{Tifinagh} NOR \p{Is_Tifinagh}) (80)</code>
<code>\p{Block: Ugaritic}</code>	<code>(Single: \p{InUgaritic}; NOT \p{Ugaritic} NOR \p{Is_Ugaritic}) (32)</code>
<code>\p{Block: Unified_Canadian_Aboriginal_Syllabics}</code>	<code>(Short: \p{Blk=CanadianSyllabics}, \p{InCanadianSyllabics}) (640)</code>
<code>\p{Block: Unified_Canadian_Aboriginal_Syllabics_Extended}</code>	<code>(Single: \p{InUnifiedCanadianAboriginalSyllabicsExtended}) (80)</code>
<code>\p{Block: Vai}</code>	<code>(Single: \p{InVai}; NOT \p{Vai} NOR \p{Is_Vai}) (320)</code>
<code>\p{Block: Variation_Selectors}</code>	<code>(Single: \p{InVariationSelectors}) (16)</code>
<code>\p{Block: Variation_Selectors_Supplement}</code>	<code>(Single: \p{InVariationSelectorsSupplement}) (240)</code>
<code>\p{Block: Vedic_Extensions}</code>	<code>(Single: \p{InVedicExtensions}) (48)</code>
<code>\p{Block: Vertical_Forms}</code>	<code>(Single: \p{InVerticalForms}) (16)</code>
<code>\p{Block: Yi_Radicals}</code>	<code>(Single: \p{InYiRadicals}) (64)</code>
<code>\p{Block: Yi_Syllables}</code>	<code>(Single: \p{InYiSyllables}) (1168)</code>
<code>\p{Block: Yijing_Hexagram_Symbols}</code>	<code>(Single: \p{InYijingHexagramSymbols}) (64)</code>
X <code>\p{Block_Elements}</code>	<code>\p{Block=Block_Elements} (32)</code>
<code>\p{Bopo}</code>	<code>\p{Bopomofo} (= \p{Script=Bopomofo}) (NOT \p{Block=Bopomofo}) (65)</code>
<code>\p{Bopomofo}</code>	<code>\p{Script=Bopomofo} (Short: \p{Bopo}; NOT \p{Block=Bopomofo}) (65)</code>
X <code>\p{Bopomofo_Extended}</code>	<code>\p{Block=Bopomofo_Extended} (32)</code>
X <code>\p{Box_Drawing}</code>	<code>\p{Block=Box_Drawing} (128)</code>
<code>\p{Brai}</code>	<code>\p{Braille} (= \p{Script=Braille}) (256)</code>
<code>\p{Braille}</code>	<code>\p{Script=Braille} (Short: \p{Brai}) (256)</code>
X <code>\p{Braille_Patterns}</code>	<code>\p{Block=Braille_Patterns} (256)</code>
<code>\p{Bugi}</code>	<code>\p{Buginese} (= \p{Script=Buginese}) (NOT \p{Block=Buginese}) (30)</code>
<code>\p{Buginese}</code>	<code>\p{Script=Buginese} (Short: \p{Bugi}; NOT \p{Block=Buginese}) (30)</code>
<code>\p{Buhd}</code>	<code>\p{Buhid} (= \p{Script=Buhid}) (NOT \p{Block=Buhid}) (20)</code>
<code>\p{Buhid}</code>	<code>\p{Script=Buhid} (Short: \p{Buhd}; NOT \p{Block=Buhid}) (20)</code>
X <code>\p{Byzantine_Musical_Symbols}</code>	<code>\p{Block=Byzantine_Musical_Symbols} (256)</code>
<code>\p{C}</code>	<code>\p{Other} (= \p{General_Category=Other}) (1_006_956)</code>
<code>\p{Canadian_Aboriginal}</code>	<code>\p{Script=Canadian_Aboriginal} (Short: \p{Cans}) (710)</code>
X <code>\p{Canadian_Syllabics}</code>	<code>\p{Unified_Canadian_Aboriginal_Syllabics} (= \p{Block=Unified_Canadian_Aboriginal_Syllabics}) (640)</code>

```

T \p{Canonical_Combining_Class: 0} \p{Canonical_Combining_Class=
    Not_Reordered} (1_113_518)
T \p{Canonical_Combining_Class: 1} \p{Canonical_Combining_Class=
    Overlay} (26)
T \p{Canonical_Combining_Class: 7} \p{Canonical_Combining_Class=
    Nukta} (11)
T \p{Canonical_Combining_Class: 8} \p{Canonical_Combining_Class=
    Kana_Voicing} (2)
T \p{Canonical_Combining_Class: 9} \p{Canonical_Combining_Class=
    Virama} (27)
T \p{Canonical_Combining_Class: 10} (Short: \p{Ccc=10}) (1)
T \p{Canonical_Combining_Class: 11} (Short: \p{Ccc=11}) (1)
T \p{Canonical_Combining_Class: 12} (Short: \p{Ccc=12}) (1)
T \p{Canonical_Combining_Class: 13} (Short: \p{Ccc=13}) (1)
T \p{Canonical_Combining_Class: 14} (Short: \p{Ccc=14}) (1)
T \p{Canonical_Combining_Class: 15} (Short: \p{Ccc=15}) (1)
T \p{Canonical_Combining_Class: 16} (Short: \p{Ccc=16}) (1)
T \p{Canonical_Combining_Class: 17} (Short: \p{Ccc=17}) (1)
T \p{Canonical_Combining_Class: 18} (Short: \p{Ccc=18}) (2)
T \p{Canonical_Combining_Class: 19} (Short: \p{Ccc=19}) (2)
T \p{Canonical_Combining_Class: 20} (Short: \p{Ccc=20}) (1)
T \p{Canonical_Combining_Class: 21} (Short: \p{Ccc=21}) (1)
T \p{Canonical_Combining_Class: 22} (Short: \p{Ccc=22}) (1)
T \p{Canonical_Combining_Class: 23} (Short: \p{Ccc=23}) (1)
T \p{Canonical_Combining_Class: 24} (Short: \p{Ccc=24}) (1)
T \p{Canonical_Combining_Class: 25} (Short: \p{Ccc=25}) (1)
T \p{Canonical_Combining_Class: 26} (Short: \p{Ccc=26}) (1)
T \p{Canonical_Combining_Class: 27} (Short: \p{Ccc=27}) (1)
T \p{Canonical_Combining_Class: 28} (Short: \p{Ccc=28}) (1)
T \p{Canonical_Combining_Class: 29} (Short: \p{Ccc=29}) (1)
T \p{Canonical_Combining_Class: 30} (Short: \p{Ccc=30}) (2)
T \p{Canonical_Combining_Class: 31} (Short: \p{Ccc=31}) (2)
T \p{Canonical_Combining_Class: 32} (Short: \p{Ccc=32}) (2)
T \p{Canonical_Combining_Class: 33} (Short: \p{Ccc=33}) (1)
T \p{Canonical_Combining_Class: 34} (Short: \p{Ccc=34}) (1)
T \p{Canonical_Combining_Class: 35} (Short: \p{Ccc=35}) (1)
T \p{Canonical_Combining_Class: 36} (Short: \p{Ccc=36}) (1)
T \p{Canonical_Combining_Class: 84} (Short: \p{Ccc=84}) (1)
T \p{Canonical_Combining_Class: 91} (Short: \p{Ccc=91}) (1)
T \p{Canonical_Combining_Class: 103} (Short: \p{Ccc=103}) (2)
T \p{Canonical_Combining_Class: 107} (Short: \p{Ccc=107}) (4)
T \p{Canonical_Combining_Class: 118} (Short: \p{Ccc=118}) (2)
T \p{Canonical_Combining_Class: 122} (Short: \p{Ccc=122}) (4)
T \p{Canonical_Combining_Class: 129} (Short: \p{Ccc=129}) (1)
T \p{Canonical_Combining_Class: 130} (Short: \p{Ccc=130}) (6)
T \p{Canonical_Combining_Class: 132} (Short: \p{Ccc=132}) (1)
T \p{Canonical_Combining_Class: 200} \p{Canonical_Combining_Class=
    Attached_Below_Left} (0)
T \p{Canonical_Combining_Class: 202} \p{Canonical_Combining_Class=
    Attached_Below} (5)
T \p{Canonical_Combining_Class: 214} \p{Canonical_Combining_Class=
    Attached_Above} (1)
T \p{Canonical_Combining_Class: 216} \p{Canonical_Combining_Class=
    Attached_Above_Right} (9)
T \p{Canonical_Combining_Class: 218} \p{Canonical_Combining_Class=
    Below_Left} (1)

```

```

T \p{Canonical_Combining_Class: 220} \p{Canonical_Combining_Class=
    Below} (117)
T \p{Canonical_Combining_Class: 222} \p{Canonical_Combining_Class=
    Below_Right} (4)
T \p{Canonical_Combining_Class: 224} \p{Canonical_Combining_Class=
    Left} (2)
T \p{Canonical_Combining_Class: 226} \p{Canonical_Combining_Class=
    Right} (1)
T \p{Canonical_Combining_Class: 228} \p{Canonical_Combining_Class=
    Above_Left} (3)
T \p{Canonical_Combining_Class: 230} \p{Canonical_Combining_Class=
    Above} (318)
T \p{Canonical_Combining_Class: 232} \p{Canonical_Combining_Class=
    Above_Right} (4)
T \p{Canonical_Combining_Class: 233} \p{Canonical_Combining_Class=
    Double_Below} (3)
T \p{Canonical_Combining_Class: 234} \p{Canonical_Combining_Class=
    Double_Above} (5)
T \p{Canonical_Combining_Class: 240} \p{Canonical_Combining_Class=
    Iota_Subscript} (1)
\p{Canonical_Combining_Class: A} \p{Canonical_Combining_Class=
    Above} (318)
\p{Canonical_Combining_Class: Above} (Short: \p{Ccc=A}) (318)
\p{Canonical_Combining_Class: Above_Left} (Short: \p{Ccc=AL}) (3)
\p{Canonical_Combining_Class: Above_Right} (Short: \p{Ccc=AR}) (4)
\p{Canonical_Combining_Class: AL} \p{Canonical_Combining_Class=
    Above_Left} (3)
\p{Canonical_Combining_Class: AR} \p{Canonical_Combining_Class=
    Above_Right} (4)
\p{Canonical_Combining_Class: ATA} \p{Canonical_Combining_Class=
    Attached_Above} (1)
\p{Canonical_Combining_Class: ATAR} \p{Canonical_Combining_Class=
    Attached_Above_Right} (9)
\p{Canonical_Combining_Class: ATB} \p{Canonical_Combining_Class=
    Attached_Below} (5)
\p{Canonical_Combining_Class: ATBL} \p{Canonical_Combining_Class=
    Attached_Below_Left} (0)
\p{Canonical_Combining_Class: Attached_Above} (Short: \p{Ccc=ATA})
    (1)
\p{Canonical_Combining_Class: Attached_Above_Right} (Short:
    \p{Ccc=ATAR}) (9)
\p{Canonical_Combining_Class: Attached_Below} (Short: \p{Ccc=ATB})
    (5)
\p{Canonical_Combining_Class: Attached_Below_Left} (Short: \p{Ccc=
    ATBL}) (0)
\p{Canonical_Combining_Class: B} \p{Canonical_Combining_Class=
    Below} (117)
\p{Canonical_Combining_Class: Below} (Short: \p{Ccc=B}) (117)
\p{Canonical_Combining_Class: Below_Left} (Short: \p{Ccc=BL}) (1)
\p{Canonical_Combining_Class: Below_Right} (Short: \p{Ccc=BR}) (4)
\p{Canonical_Combining_Class: BL} \p{Canonical_Combining_Class=
    Below_Left} (1)
\p{Canonical_Combining_Class: BR} \p{Canonical_Combining_Class=
    Below_Right} (4)
\p{Canonical_Combining_Class: DA} \p{Canonical_Combining_Class=
    Double_Above} (5)

```

```

\p{Canonical_Combining_Class: DB} \p{Canonical_Combining_Class=
    Double_Below} (3)
\p{Canonical_Combining_Class: Double_Above} (Short: \p{Ccc=DA}) (5)
\p{Canonical_Combining_Class: Double_Below} (Short: \p{Ccc=DB}) (3)
\p{Canonical_Combining_Class: Iota_Subscript} (Short: \p{Ccc=IS})
    (1)
\p{Canonical_Combining_Class: IS} \p{Canonical_Combining_Class=
    Iota_Subscript} (1)
\p{Canonical_Combining_Class: Kana_Voicing} (Short: \p{Ccc=KV}) (2)
\p{Canonical_Combining_Class: KV} \p{Canonical_Combining_Class=
    Kana_Voicing} (2)
\p{Canonical_Combining_Class: L} \p{Canonical_Combining_Class=
    Left} (2)
\p{Canonical_Combining_Class: Left} (Short: \p{Ccc=L}) (2)
\p{Canonical_Combining_Class: NK} \p{Canonical_Combining_Class=
    Nukta} (11)
\p{Canonical_Combining_Class: Not_Reordered} (Short: \p{Ccc=NR})
    (1_113_518)
\p{Canonical_Combining_Class: NR} \p{Canonical_Combining_Class=
    Not_Reordered} (1_113_518)
\p{Canonical_Combining_Class: Nukta} (Short: \p{Ccc=NK}) (11)
\p{Canonical_Combining_Class: OV} \p{Canonical_Combining_Class=
    Overlay} (26)
\p{Canonical_Combining_Class: Overlay} (Short: \p{Ccc=OV}) (26)
\p{Canonical_Combining_Class: R} \p{Canonical_Combining_Class=
    Right} (1)
\p{Canonical_Combining_Class: Right} (Short: \p{Ccc=R}) (1)
\p{Canonical_Combining_Class: Virama} (Short: \p{Ccc=VR}) (27)
\p{Canonical_Combining_Class: VR} \p{Canonical_Combining_Class=
    Virama} (27)
\p{Cans} \p{Canadian_Aboriginal} (= \p{Script=
    Canadian_Aboriginal}) (710)
\p{Cari} \p{Carian} (= \p{Script=Carian}) (NOT
    \p{Block=Carian}) (49)
\p{Carian} \p{Script=Carian} (Short: \p{Cari}; NOT
    \p{Block=Carian}) (49)
\p{Case_Ignorable} \p{Case_Ignorable=Y} (Short: \p{CI}) (1632)
\p{Case_Ignorable: N*} (Short: \p{CI=N}, \p{CI}) (1_112_480)
\p{Case_Ignorable: Y*} (Short: \p{CI=Y}, \p{CI}) (1632)
\p{Cased} \p{Cased=Y} (3408)
\p{Cased: N*} (Single: \p{Cased}) (1_110_704)
\p{Cased: Y*} (Single: \p{Cased}) (3408)
\p{Cased_Letter} \p{General_Category=Cased_Letter} (Short:
    \p{LC}) (3207)
\p{Category: *} \p{General_Category: *}
\p{Cc} \p{Cntrl} (= \p{General_Category=Control})
    (65)
\p{Ccc: *} \p{Canonical_Combining_Class: *}
\p{CE} \p{Composition_Exclusion} (=
    \p{Composition_Exclusion=Y}) (81)
\p{CE: *} \p{Composition_Exclusion: *}
\p{Cf} \p{Format} (= \p{General_Category=Format})
    (140)
\p{Cham} \p{Script=Cham} (NOT \p{Block=Cham}) (83)
\p{Changes_When_Casefolded} \p{Changes_When_Casefolded=Y} (Short:
    \p{CWCFF}) (1093)
    
```



```

\p{Changes_When_Casefolded: N*} (Short: \p{CWCF=N}, \P{CWCF})
    (1_113_019)
\p{Changes_When_Casefolded: Y*} (Short: \p{CWCF=Y}, \P{CWCF})
    (1093)
\p{Changes_When_Casemapped} \p{Changes_When_Casemapped=Y} (Short:
    \p{CWCM}) (2110)
\p{Changes_When_Casemapped: N*} (Short: \p{CWCM=N}, \P{CWCM})
    (1_112_002)
\p{Changes_When_Casemapped: Y*} (Short: \p{CWCM=Y}, \P{CWCM})
    (2110)
\p{Changes_When_Lowercased} \p{Changes_When_Lowercased=Y} (Short:
    \p{CWL}) (1029)
\p{Changes_When_Lowercased: N*} (Short: \p{CWL=N}, \P{CWL})
    (1_113_083)
\p{Changes_When_Lowercased: Y*} (Short: \p{CWL=Y}, \P{CWL}) (1029)
\p{Changes_When_NFKC_Casefolded} \p{Changes_When_NFKC_Casefolded=
    Y} (Short: \p{CWKCF}) (9740)
\p{Changes_When_NFKC_Casefolded: N*} (Short: \p{CWKCF=N},
    \P{CWKCF}) (1_104_372)
\p{Changes_When_NFKC_Casefolded: Y*} (Short: \p{CWKCF=Y},
    \P{CWKCF}) (9740)
\p{Changes_When_Titlecased} \p{Changes_When_Titlecased=Y} (Short:
    \p{CWT}) (1085)
\p{Changes_When_Titlecased: N*} (Short: \p{CWT=N}, \P{CWT})
    (1_113_027)
\p{Changes_When_Titlecased: Y*} (Short: \p{CWT=Y}, \P{CWT}) (1085)
\p{Changes_When_Uppercased} \p{Changes_When_Uppercased=Y} (Short:
    \p{CWU}) (1112)
\p{Changes_When_Uppercased: N*} (Short: \p{CWU=N}, \P{CWU})
    (1_113_000)
\p{Changes_When_Uppercased: Y*} (Short: \p{CWU=Y}, \P{CWU}) (1112)
\p{Cher} \p{Cherokee} (= \p{Script=Cherokee}) (NOT
    \p{Block=Cherokee}) (85)
\p{Cherokee} \p{Script=Cherokee} (Short: \p{Cher}; NOT
    \p{Block=Cherokee}) (85)
\p{CI} \p{Case_Ignorable} (= \p{Case_Ignorable=
    Y}) (1632)
\p{CI: *} \p{Case_Ignorable: *}
X \p{CJK_Compatibility} \p{Block=CJK_Compatibility} (256)
X \p{CJK_Compatibility_Forms} \p{Block=CJK_Compatibility_Forms} (32)
X \p{CJK_Compatibility_Ideographs} \p{Block=
    CJK_Compatibility_Ideographs} (512)
X \p{CJK_Compatibility_Ideographs_Supplement} \p{Block=
    CJK_Compatibility_Ideographs_Supplement}
    (544)
X \p{CJK_Radicals_Supplement} \p{Block=CJK_Radicals_Supplement} (128)
X \p{CJK_Strokes} \p{Block=CJK_Strokes} (48)
X \p{CJK_Symbols_And_Punctuation} \p{Block=
    CJK_Symbols_And_Punctuation} (64)
X \p{CJK_Unified_Ideographs} \p{Block=CJK_Unified_Ideographs}
    (20_992)
X \p{CJK_Unified_Ideographs_Extension_A} \p{Block=
    CJK_Unified_Ideographs_Extension_A}
    (6592)
X \p{CJK_Unified_Ideographs_Extension_B} \p{Block=
    CJK_Unified_Ideographs_Extension_B}

```

	(42_720)
X \p{CJK_Unified_Ideographs_Extension_C}	\p{Block=CJK_Unified_Ideographs_Extension_C} (4160)
\p{Close_Punctuation}	\p{General_Category=Close_Punctuation} (Short: \p{Pe}) (71)
\p{Cn}	\p{Unassigned} (= \p{General_Category=Unassigned}) (867_235)
\p{Cntrl}	\p{General_Category=Control} Control characters (Short: \p{Cc}) (65)
\p{Co}	\p{Private_Use} (= \p{General_Category=Private_Use}) (NOT \p{Private_Use_Area}) (137_468)
X \p{Combining_Diacritical_Marks}	\p{Block=Combining_Diacritical_Marks} (112)
X \p{Combining_Diacritical_Marks_For_Symbols}	\p{Block=Combining_Diacritical_Marks_For_Symbols} (Short: \p{InCombiningMarksForSymbols}) (48)
X \p{Combining_Diacritical_Marks_Supplement}	\p{Block=Combining_Diacritical_Marks_Supplement} (64)
X \p{Combining_Half_Marks}	\p{Block=Combining_Half_Marks} (16)
X \p{Combining_Marks_For_Symbols}	\p{Combining_Diacritical_Marks_For_Symbols} (= \p{Block=Combining_Diacritical_Marks_For_Symbols}) (48)
\p{Common}	\p{Script=Common} (Short: \p{Zyyy}) (5395)
X \p{Common_Indic_Number_Forms}	\p{Block=Common_Indic_Number_Forms} (16)
\p{Comp_Ex}	\p{Full_Composition_Exclusion} (= \p{Full_Composition_Exclusion=Y}) (1118)
\p{Comp_Ex: *}	\p{Full_Composition_Exclusion: *}
\p{Composition_Exclusion}	\p{Composition_Exclusion=Y} (Short: \p{CE}) (81)
\p{Composition_Exclusion: N*}	(Short: \p{CE=N}, \p{CE}) (1_114_031)
\p{Composition_Exclusion: Y*}	(Short: \p{CE=Y}, \p{CE}) (81)
\p{Connector_Punctuation}	\p{General_Category=Connector_Punctuation} (Short: \p{Pc}) (10)
\p{Control}	\p{Cntrl} (= \p{General_Category=Control}) (65)
X \p{Control_Pictures}	\p{Block=Control_Pictures} (64)
\p{Copt}	\p{Coptic} (= \p{Script=Coptic}) (NOT \p{Block=Coptic}) (135)
\p{Coptic}	\p{Script=Coptic} (Short: \p{Copt}; NOT \p{Block=Coptic}) (135)
X \p{Counting_Rod_Numerals}	\p{Block=Counting_Rod_Numerals} (32)
\p{Cprt}	\p{Cypriot} (= \p{Script=Cypriot}) (55)
\p{Cs}	\p{Surrogate} (= \p{General_Category=Surrogate}) (2048)
\p{Cuneiform}	\p{Script=Cuneiform} (Short: \p{Xsux}; NOT \p{Block=Cuneiform}) (982)
X \p{Cuneiform_Numbers_And_Punctuation}	\p{Block=Cuneiform_Numbers_And_Punctuation} (128)

	<code>\p{Currency_Symbol}</code>	<code>\p{General_Category=Currency_Symbol}</code> (Short: <code>\p{Sc}</code>) (46)
X	<code>\p{Currency_Symbols}</code>	<code>\p{Block=Currency_Symbols}</code> (48)
	<code>\p{CWCF}</code>	<code>\p{Changes_When_Casefolded}</code> (= <code>\p{Changes_When_Casefolded=Y}</code>) (1093)
	<code>\p{CWCF: *}</code>	<code>\p{Changes_When_Casefolded: *}</code>
	<code>\p{CWCM}</code>	<code>\p{Changes_When_Casemapped}</code> (= <code>\p{Changes_When_Casemapped=Y}</code>) (2110)
	<code>\p{CWCM: *}</code>	<code>\p{Changes_When_Casemapped: *}</code>
	<code>\p{CWKCF}</code>	<code>\p{Changes_When_NFKC_Casefolded}</code> (= <code>\p{Changes_When_NFKC_Casefolded=Y}</code>) (9740)
	<code>\p{CWKCF: *}</code>	<code>\p{Changes_When_NFKC_Casefolded: *}</code>
	<code>\p{CWL}</code>	<code>\p{Changes_When_Lowercased}</code> (= <code>\p{Changes_When_Lowercased=Y}</code>) (1029)
	<code>\p{CWL: *}</code>	<code>\p{Changes_When_Lowercased: *}</code>
	<code>\p{CWT}</code>	<code>\p{Changes_When_Titlecased}</code> (= <code>\p{Changes_When_Titlecased=Y}</code>) (1085)
	<code>\p{CWT: *}</code>	<code>\p{Changes_When_Titlecased: *}</code>
	<code>\p{CWU}</code>	<code>\p{Changes_When_Uppercased}</code> (= <code>\p{Changes_When_Uppercased=Y}</code>) (1112)
	<code>\p{CWU: *}</code>	<code>\p{Changes_When_Uppercased: *}</code>
	<code>\p{Cypriot}</code>	<code>\p{Script=Cypriot}</code> (Short: <code>\p{Cprt}</code>) (55)
X	<code>\p{Cypriot_Syllabary}</code>	<code>\p{Block=Cypriot_Syllabary}</code> (64)
	<code>\p{Cyrillic}</code>	<code>\p{Script=Cyrillic}</code> (Short: <code>\p{Cyr1}</code> ; NOT <code>\p{Block=Cyrillic}</code>) (404)
X	<code>\p{Cyrillic_Extended_A}</code>	<code>\p{Block=Cyrillic_Extended_A}</code> (32)
X	<code>\p{Cyrillic_Extended_B}</code>	<code>\p{Block=Cyrillic_Extended_B}</code> (96)
X	<code>\p{Cyrillic_Supplement}</code>	<code>\p{Block=Cyrillic_Supplement}</code> (48)
X	<code>\p{Cyrillic_Supplementary}</code>	<code>\p{Cyrillic_Supplementary}</code> (= <code>\p{Block=Cyrillic_Supplement}</code>) (48)
	<code>\p{Cyr1}</code>	<code>\p{Cyrillic}</code> (= <code>\p{Script=Cyrillic}</code>) (NOT <code>\p{Block=Cyrillic}</code>) (404)
	<code>\p{Dash}</code>	<code>\p{Dash=Y}</code> (25)
	<code>\p{Dash: N*}</code>	(Single: <code>\p{Dash}</code>) (1_114_087)
	<code>\p{Dash: Y*}</code>	(Single: <code>\p{Dash}</code>) (25)
	<code>\p{Dash_Punctuation}</code>	<code>\p{General_Category=Dash_Punctuation}</code> (Short: <code>\p{Pd}</code>) (21)
	<code>\p{Decimal_Number}</code>	<code>\p{Digit}</code> (= <code>\p{General_Category=Decimal_Number}</code>) (411)
	<code>\p{Decomposition_Type: Can}</code>	<code>\p{Decomposition_Type=Canonical}</code> (13_221)
	<code>\p{Decomposition_Type: Canonical}</code>	(Short: <code>\p{Dt=Can}</code>) (13_221)
	<code>\p{Decomposition_Type: Circle}</code>	(Short: <code>\p{Dt=Enc}</code>) (238)
	<code>\p{Decomposition_Type: Com}</code>	<code>\p{Decomposition_Type=Compat}</code> (720)
	<code>\p{Decomposition_Type: Compat}</code>	(Short: <code>\p{Dt=Com}</code>) (720)
	<code>\p{Decomposition_Type: Enc}</code>	<code>\p{Decomposition_Type=Circle}</code> (238)
	<code>\p{Decomposition_Type: Fin}</code>	<code>\p{Decomposition_Type=Final}</code> (240)
	<code>\p{Decomposition_Type: Final}</code>	(Short: <code>\p{Dt=Fin}</code>) (240)
	<code>\p{Decomposition_Type: Font}</code>	(Short: <code>\p{Dt=Font}</code>) (1043)
	<code>\p{Decomposition_Type: Fra}</code>	<code>\p{Decomposition_Type=Fraction}</code> (20)
	<code>\p{Decomposition_Type: Fraction}</code>	(Short: <code>\p{Dt=Fra}</code>) (20)
	<code>\p{Decomposition_Type: Init}</code>	<code>\p{Decomposition_Type=Initial}</code> (171)
	<code>\p{Decomposition_Type: Initial}</code>	(Short: <code>\p{Dt=Init}</code>) (171)
	<code>\p{Decomposition_Type: Iso}</code>	<code>\p{Decomposition_Type=Isolated}</code> (238)
	<code>\p{Decomposition_Type: Isolated}</code>	(Short: <code>\p{Dt=Iso}</code>) (238)

<code>\p{Decomposition_Type: Med}</code>	<code>\p{Decomposition_Type=Medial}</code>	(82)
<code>\p{Decomposition_Type: Medial}</code>	(Short: <code>\p{Dt=Med}</code>)	(82)
<code>\p{Decomposition_Type: Nar}</code>	<code>\p{Decomposition_Type=Narrow}</code>	(122)
<code>\p{Decomposition_Type: Narrow}</code>	(Short: <code>\p{Dt=Nar}</code>)	(122)
<code>\p{Decomposition_Type: Nb}</code>	<code>\p{Decomposition_Type=Nobreak}</code>	(5)
<code>\p{Decomposition_Type: Nobreak}</code>	(Short: <code>\p{Dt=Nb}</code>)	(5)
<code>\p{Decomposition_Type: Non_Canon}</code>	<code>\p{Decomposition_Type=Non_Canonical}</code>	(Perl extension) (3467)
<code>\p{Decomposition_Type: Non_Canonical}</code>	Union of all non-canonical decompositions (Short: <code>\p{Dt=NonCanon}</code>)	(Perl extension) (3467)
<code>\p{Decomposition_Type: None}</code>	(Short: <code>\p{Dt=None}</code>)	(1_097_424)
<code>\p{Decomposition_Type: Small}</code>	(Short: <code>\p{Dt=Sml}</code>)	(26)
<code>\p{Decomposition_Type: Sml}</code>	<code>\p{Decomposition_Type=Small}</code>	(26)
<code>\p{Decomposition_Type: Sqr}</code>	<code>\p{Decomposition_Type=Square}</code>	(251)
<code>\p{Decomposition_Type: Square}</code>	(Short: <code>\p{Dt=Sqr}</code>)	(251)
<code>\p{Decomposition_Type: Sub}</code>	(Short: <code>\p{Dt=Sub}</code>)	(30)
<code>\p{Decomposition_Type: Sup}</code>	<code>\p{Decomposition_Type=Super}</code>	(142)
<code>\p{Decomposition_Type: Super}</code>	(Short: <code>\p{Dt=Sup}</code>)	(142)
<code>\p{Decomposition_Type: Vert}</code>	<code>\p{Decomposition_Type=Vertical}</code>	(35)
<code>\p{Decomposition_Type: Vertical}</code>	(Short: <code>\p{Dt=Vert}</code>)	(35)
<code>\p{Decomposition_Type: Wide}</code>	(Short: <code>\p{Dt=Wide}</code>)	(104)
<code>\p{Default_Ignorable_Code_Point}</code>	<code>\p{Default_Ignorable_Code_Point=Y}</code>	(Short: <code>\p{DI}</code>) (4167)
<code>\p{Default_Ignorable_Code_Point: N*}</code>	(Short: <code>\p{DI=N}</code> , <code>\P{DI}</code>)	(1_109_945)
<code>\p{Default_Ignorable_Code_Point: Y*}</code>	(Short: <code>\p{DI=Y}</code> , <code>\P{DI}</code>)	(4167)
<code>\p{Dep}</code>	<code>\p{Deprecated}</code>	(= <code>\p{Deprecated=Y}</code>) (110)
<code>\p{Dep: *}</code>	<code>\p{Deprecated: *}</code>	
<code>\p{Deprecated}</code>	<code>\p{Deprecated=Y}</code>	(Short: <code>\p{Dep}</code>) (110)
<code>\p{Deprecated: N*}</code>	(Short: <code>\p{Dep=N}</code> , <code>\P{Dep}</code>)	(1_114_002)
<code>\p{Deprecated: Y*}</code>	(Short: <code>\p{Dep=Y}</code> , <code>\P{Dep}</code>)	(110)
<code>\p{Deseret}</code>	<code>\p{Script=Deseret}</code>	(Short: <code>\p{Dsrt}</code>) (80)
<code>\p{Deva}</code>	<code>\p{Devanagari}</code>	(= <code>\p{Script=Devanagari}</code>) (NOT <code>\p{Block=Devanagari}</code>) (140)
<code>\p{Devanagari}</code>	<code>\p{Script=Devanagari}</code>	(Short: <code>\p{Deva}</code> ; NOT <code>\p{Block=Devanagari}</code>) (140)
X <code>\p{Devanagari_Extended}</code>	<code>\p{Block=Devanagari_Extended}</code>	(32)
<code>\p{DI}</code>	<code>\p{Default_Ignorable_Code_Point}</code>	(= <code>\p{Default_Ignorable_Code_Point=Y}</code>) (4167)
<code>\p{DI: *}</code>	<code>\p{Default_Ignorable_Code_Point: *}</code>	
<code>\p{Dia}</code>	<code>\p{Diacritic}</code>	(= <code>\p{Diacritic=Y}</code>) (639)
<code>\p{Dia: *}</code>	<code>\p{Diacritic: *}</code>	
<code>\p{Diacritic}</code>	<code>\p{Diacritic=Y}</code>	(Short: <code>\p{Dia}</code>) (639)
<code>\p{Diacritic: N*}</code>	(Short: <code>\p{Dia=N}</code> , <code>\P{Dia}</code>)	(1_113_473)
<code>\p{Diacritic: Y*}</code>	(Short: <code>\p{Dia=Y}</code> , <code>\P{Dia}</code>)	(639)
<code>\p{Digit}</code>	<code>\p{General_Category=Decimal_Number}</code>	<code>\d</code> , extended beyond just [0-9] (Short: <code>\p{Nd}</code>) (411)
X <code>\p{Dingbats}</code>	<code>\p{Block=Dingbats}</code>	(192)
X <code>\p{Domino_Tiles}</code>	<code>\p{Block=Domino_Tiles}</code>	(112)
<code>\p{Dsrt}</code>	<code>\p{Deseret}</code>	(= <code>\p{Script=Deseret}</code>) (80)
<code>\p{Dt: *}</code>	<code>\p{Decomposition_Type: *}</code>	
<code>\p{Ea: *}</code>	<code>\p{East_Asian_Width: *}</code>	

```

\p{East_Asian_Width: A} \p{East_Asian_Width=Ambiguous} (138_666)
\p{East_Asian_Width: Ambiguous} (Short: \p{Ea=A}) (138_666)
\p{East_Asian_Width: F} \p{East_Asian_Width=Fullwidth} (104)
\p{East_Asian_Width: Fullwidth} (Short: \p{Ea=F}) (104)
\p{East_Asian_Width: H} \p{East_Asian_Width=Halfwidth} (123)
\p{East_Asian_Width: Halfwidth} (Short: \p{Ea=H}) (123)
\p{East_Asian_Width: N} \p{East_Asian_Width=Neutral} (801_909)
\p{East_Asian_Width: Na} \p{East_Asian_Width=Narrow} (111)
\p{East_Asian_Width: Narrow} (Short: \p{Ea=Na}) (111)
\p{East_Asian_Width: Neutral} (Short: \p{Ea=N}) (801_909)
\p{East_Asian_Width: W} \p{East_Asian_Width=Wide} (173_199)
\p{East_Asian_Width: Wide} (Short: \p{Ea=W}) (173_199)
\p{Egyp} \p{Egyptian_Hieroglyphs} (= \p{Script=
Egyptian_Hieroglyphs}) (NOT \p{Block=
Egyptian_Hieroglyphs}) (1071)
\p{Egyptian_Hieroglyphs} \p{Script=Egyptian_Hieroglyphs} (Short:
\p{Egyp}; NOT \p{Block=
Egyptian_Hieroglyphs}) (1071)
X \p{Enclosed_Alphanumeric_Supplement} \p{Block=
Enclosed_Alphanumeric_Supplement} (256)
X \p{Enclosed_Alphanumerics} \p{Block=Enclosed_Alphanumerics} (160)
X \p{Enclosed_CJK_Letters_And_Months} \p{Block=
Enclosed_CJK_Letters_And_Months} (256)
X \p{Enclosed_Ideographic_Supplement} \p{Block=
Enclosed_Ideographic_Supplement} (256)
\p{Enclosing_Mark} \p{General_Category=Enclosing_Mark}
(Short: \p{Me}) (13)
\p{Ethi} \p{Ethiopic} (= \p{Script=Ethiopic}) (NOT
\p{Block=Ethiopic}) (461)
\p{Ethiopic} \p{Script=Ethiopic} (Short: \p{Ethi}; NOT
\p{Block=Ethiopic}) (461)
X \p{Ethiopic_Extended} \p{Block=Ethiopic_Extended} (96)
X \p{Ethiopic_Supplement} \p{Block=Ethiopic_Supplement} (32)
\p{Ext} \p{Extender} (= \p{Extender=Y}) (28)
\p{Ext: *} \p{Extender: *}
\p{Extender} \p{Extender=Y} (Short: \p{Ext}) (28)
\p{Extender: N*} (Short: \p{Ext=N}, \p{Ext}) (1_114_084)
\p{Extender: Y*} (Short: \p{Ext=Y}, \p{Ext}) (28)
\p{Final_Punctuation} \p{General_Category=Final_Punctuation}
(Short: \p{Pf}) (10)
\p{Format} \p{General_Category=Format} (Short:
\p{Cf}) (140)
\p{Full_Composition_Exclusion} \p{Full_Composition_Exclusion=Y}
(Short: \p{CompEx}) (1118)
\p{Full_Composition_Exclusion: N*} (Short: \p{CompEx=N},
\p{CompEx}) (1_112_994)
\p{Full_Composition_Exclusion: Y*} (Short: \p{CompEx=Y},
\p{CompEx}) (1118)
\p{Gc: *} \p{General_Category: *}
\p{GCB: *} \p{Grapheme_Cluster_Break: *}
\p{General_Category: C} \p{General_Category=Other} (1_006_956)
\p{General_Category: Cased_Letter} [\p{Ll}\p{Lu}\p{Lt}] (Short:
\p{Gc=LC}, \p{LC}) (3207)
\p{General_Category: Cc} \p{General_Category=Control} (65)
\p{General_Category: Cf} \p{General_Category=Format} (140)
\p{General_Category: Close_Punctuation} (Short: \p{Gc=Pe}, \p{Pe})

```

```

(71)
\p{General_Category: Cn} \p{General_Category=Unassigned} (867_235)
\p{General_Category: Cntrl} \p{General_Category=Control} (65)
\p{General_Category: Co} \p{General_Category=Private_Use} (137_468)
\p{General_Category: Connector_Punctuation} (Short: \p{Gc=Pc},
    \p{Pc}) (10)
\p{General_Category: Control} (Short: \p{Gc=Cc}, \p{Cc}) (65)
\p{General_Category: Cs} \p{General_Category=Surrogate} (2048)
\p{General_Category: Currency_Symbol} (Short: \p{Gc=Sc}, \p{Sc})
    (46)
\p{General_Category: Dash_Punctuation} (Short: \p{Gc=Pd}, \p{Pd})
    (21)
\p{General_Category: Decimal_Number} (Short: \p{Gc=Nd}, \p{Nd})
    (411)
\p{General_Category: Digit} \p{General_Category=Decimal_Number}
    (411)
\p{General_Category: Enclosing_Mark} (Short: \p{Gc=Me}, \p{Me})
    (13)
\p{General_Category: Final_Punctuation} (Short: \p{Gc=Pf}, \p{Pf})
    (10)
\p{General_Category: Format} (Short: \p{Gc=Cf}, \p{Cf}) (140)
\p{General_Category: Initial_Punctuation} (Short: \p{Gc=Pi},
    \p{Pi}) (12)
\p{General_Category: L} \p{General_Category=Letter} (99_537)
X \p{General_Category: L&} \p{General_Category=Cased_Letter} (3207)
X \p{General_Category: L_} \p{General_Category=Cased_Letter} (3207)
\p{General_Category: LC} \p{General_Category=Cased_Letter} (3207)
\p{General_Category: Letter} (Short: \p{Gc=L}, \p{L}) (99_537)
\p{General_Category: Letter_Number} (Short: \p{Gc=Nl}, \p{Nl})
    (224)
\p{General_Category: Line_Separator} (Short: \p{Gc=Zl}, \p{Zl}) (1)
\p{General_Category: Ll} \p{General_Category=Lowercase_Letter}
    (1749)
\p{General_Category: Lm} \p{General_Category=Modifier_Letter} (202)
\p{General_Category: Lo} \p{General_Category=Other_Letter} (96_128)
\p{General_Category: Lowercase_Letter} (Short: \p{Gc=Ll}, \p{Ll})
    (1749)
\p{General_Category: Lt} \p{General_Category=Titlecase_Letter} (31)
\p{General_Category: Lu} \p{General_Category=Uppercase_Letter}
    (1427)
\p{General_Category: M} \p{General_Category=Mark} (1451)
\p{General_Category: Mark} (Short: \p{Gc=M}, \p{M}) (1451)
\p{General_Category: Math_Symbol} (Short: \p{Gc=Sm}, \p{Sm}) (945)
\p{General_Category: Mc} \p{General_Category=Spacing_Mark} (276)
\p{General_Category: Me} \p{General_Category=Enclosing_Mark} (13)
\p{General_Category: Mn} \p{General_Category=Nonspacing_Mark}
    (1162)
\p{General_Category: Modifier_Letter} (Short: \p{Gc=Lm}, \p{Lm})
    (202)
\p{General_Category: Modifier_Symbol} (Short: \p{Gc=Sk}, \p{Sk})
    (99)
\p{General_Category: N} \p{General_Category=Number} (1064)
\p{General_Category: Nd} \p{General_Category=Decimal_Number} (411)
\p{General_Category: Nl} \p{General_Category=Letter_Number} (224)
\p{General_Category: No} \p{General_Category=Other_Number} (429)
\p{General_Category: Nonspacing_Mark} (Short: \p{Gc=Mn}, \p{Mn})

```

```

(1162)
\p{General_Category: Number} (Short: \p{Gc=N}, \p{N}) (1064)
\p{General_Category: Open_Punctuation} (Short: \p{Gc=Ps}, \p{Ps})
(72)
\p{General_Category: Other} (Short: \p{Gc=C}, \p{C}) (1_006_956)
\p{General_Category: Other_Letter} (Short: \p{Gc=Lo}, \p{Lo})
(96_128)
\p{General_Category: Other_Number} (Short: \p{Gc=No}, \p{No}) (429)
\p{General_Category: Other_Punctuation} (Short: \p{Gc=Po}, \p{Po})
(389)
\p{General_Category: Other_Symbol} (Short: \p{Gc=So}, \p{So})
(3409)
\p{General_Category: P} \p{General_Category=Punctuation} (585)
\p{General_Category: Paragraph_Separator} (Short: \p{Gc=Zp},
\p{Zp}) (1)
\p{General_Category: Pc} \p{General_Category=
Connector_Punctuation} (10)
\p{General_Category: Pd} \p{General_Category=Dash_Punctuation} (21)
\p{General_Category: Pe} \p{General_Category=Close_Punctuation}
(71)
\p{General_Category: Pf} \p{General_Category=Final_Punctuation}
(10)
\p{General_Category: Pi} \p{General_Category=Initial_Punctuation}
(12)
\p{General_Category: Po} \p{General_Category=Other_Punctuation}
(389)
\p{General_Category: Private_Use} (Short: \p{Gc=Co}, \p{Co})
(137_468)
\p{General_Category: Ps} \p{General_Category=Open_Punctuation} (72)
\p{General_Category: Punct} \p{General_Category=Punctuation} (585)
\p{General_Category: Punctuation} (Short: \p{Gc=P}, \p{P}) (585)
\p{General_Category: S} \p{General_Category=Symbol} (4499)
\p{General_Category: Sc} \p{General_Category=Currency_Symbol} (46)
\p{General_Category: Separator} (Short: \p{Gc=Z}, \p{Z}) (20)
\p{General_Category: Sk} \p{General_Category=Modifier_Symbol} (99)
\p{General_Category: Sm} \p{General_Category=Math_Symbol} (945)
\p{General_Category: So} \p{General_Category=Other_Symbol} (3409)
\p{General_Category: Space_Separator} (Short: \p{Gc=Zs}, \p{Zs})
(18)
\p{General_Category: Spacing_Mark} (Short: \p{Gc=Mc}, \p{Mc}) (276)
\p{General_Category: Surrogate} Mostly not usable in Perl. (Short:
\p{Gc=Cs}, \p{Cs}) (2048)
\p{General_Category: Symbol} (Short: \p{Gc=S}, \p{S}) (4499)
\p{General_Category: Titlecase_Letter} (Short: \p{Gc=Lt}, \p{Lt})
(31)
\p{General_Category: Unassigned} (Short: \p{Gc=Cn}, \p{Cn})
(867_235)
\p{General_Category: Uppercase_Letter} (Short: \p{Gc=Lu}, \p{Lu})
(1427)
\p{General_Category: Z} \p{General_Category=Separator} (20)
\p{General_Category: Zl} \p{General_Category=Line_Separator} (1)
\p{General_Category: Zp} \p{General_Category=Paragraph_Separator}
(1)
\p{General_Category: Zs} \p{General_Category=Space_Separator} (18)
X \p{General_Punctuation} \p{Block=General_Punctuation} (112)
X \p{Geometric_Shapes} \p{Block=Geometric_Shapes} (96)

```

<code>\p{Geor}</code>	<code>\p{Georgian}</code> (= <code>\p{Script=Georgian}</code>) (NOT <code>\p{Block=Georgian}</code>) (120)
<code>\p{Georgian}</code>	<code>\p{Script=Georgian}</code> (Short: <code>\p{Geor}</code> ; NOT <code>\p{Block=Georgian}</code>) (120)
X <code>\p{Georgian_Supplement}</code>	<code>\p{Block=Georgian_Supplement}</code> (48)
<code>\p{Glag}</code>	<code>\p{Glagolitic}</code> (= <code>\p{Script=Glagolitic}</code>) (NOT <code>\p{Block=Glagolitic}</code>) (94)
<code>\p{Glagolitic}</code>	<code>\p{Script=Glagolitic}</code> (Short: <code>\p{Glag}</code> ; NOT <code>\p{Block=Glagolitic}</code>) (94)
<code>\p{Goth}</code>	<code>\p{Gothic}</code> (= <code>\p{Script=Gothic}</code>) (NOT <code>\p{Block=Gothic}</code>) (27)
<code>\p{Gothic}</code>	<code>\p{Script=Gothic}</code> (Short: <code>\p{Goth}</code> ; NOT <code>\p{Block=Gothic}</code>) (27)
<code>\p{Gr_Base}</code>	<code>\p{Grapheme_Base}</code> (= <code>\p{Grapheme_Base=Y}</code>) (105_958)
<code>\p{Gr_Base: *}</code>	<code>\p{Grapheme_Base: *}</code>
<code>\p{Gr_Ext}</code>	<code>\p{Grapheme_Extend}</code> (= <code>\p{Grapheme_Extend=Y}</code>) (1198)
<code>\p{Gr_Ext: *}</code>	<code>\p{Grapheme_Extend: *}</code>
<code>\p{Graph}</code>	Characters that are graphical (244_744)
<code>\p{Grapheme_Base}</code>	<code>\p{Grapheme_Base=Y}</code> (Short: <code>\p{GrBase}</code>) (105_958)
<code>\p{Grapheme_Base: N*}</code>	(Short: <code>\p{GrBase=N}</code> , <code>\p{GrBase}</code>) (1_008_154)
<code>\p{Grapheme_Base: Y*}</code>	(Short: <code>\p{GrBase=Y}</code> , <code>\p{GrBase}</code>) (105_958)
<code>\p{Grapheme_Cluster_Break: CN}</code>	<code>\p{Grapheme_Cluster_Break=Control}</code> (203)
<code>\p{Grapheme_Cluster_Break: Control}</code>	(Short: <code>\p{GCB=CN}</code>) (203)
<code>\p{Grapheme_Cluster_Break: CR}</code>	(Short: <code>\p{GCB=CR}</code>) (1)
<code>\p{Grapheme_Cluster_Break: EX}</code>	<code>\p{Grapheme_Cluster_Break=Extend}</code> (1205)
<code>\p{Grapheme_Cluster_Break: Extend}</code>	(Short: <code>\p{GCB=EX}</code>) (1205)
<code>\p{Grapheme_Cluster_Break: L}</code>	(Short: <code>\p{GCB=L}</code>) (125)
<code>\p{Grapheme_Cluster_Break: LF}</code>	(Short: <code>\p{GCB=LF}</code>) (1)
<code>\p{Grapheme_Cluster_Break: LV}</code>	(Short: <code>\p{GCB=LV}</code>) (399)
<code>\p{Grapheme_Cluster_Break: LVT}</code>	(Short: <code>\p{GCB=LVT}</code>) (10_773)
<code>\p{Grapheme_Cluster_Break: Other}</code>	(Short: <code>\p{GCB=XX}</code>) (1_100_901)
<code>\p{Grapheme_Cluster_Break: PP}</code>	<code>\p{Grapheme_Cluster_Break=Prepend}</code> (15)
<code>\p{Grapheme_Cluster_Break: Prepend}</code>	(Short: <code>\p{GCB=PP}</code>) (15)
<code>\p{Grapheme_Cluster_Break: SM}</code>	<code>\p{Grapheme_Cluster_Break=SpacingMark}</code> (257)
<code>\p{Grapheme_Cluster_Break: SpacingMark}</code>	(Short: <code>\p{GCB=SM}</code>) (257)
<code>\p{Grapheme_Cluster_Break: T}</code>	(Short: <code>\p{GCB=T}</code>) (137)
<code>\p{Grapheme_Cluster_Break: V}</code>	(Short: <code>\p{GCB=V}</code>) (95)
<code>\p{Grapheme_Cluster_Break: XX}</code>	<code>\p{Grapheme_Cluster_Break=Other}</code> (1_100_901)
<code>\p{Grapheme_Extend}</code>	<code>\p{Grapheme_Extend=Y}</code> (Short: <code>\p{GrExt}</code>) (1198)
<code>\p{Grapheme_Extend: N*}</code>	(Short: <code>\p{GrExt=N}</code> , <code>\p{GrExt}</code>) (1_112_914)
<code>\p{Grapheme_Extend: Y*}</code>	(Short: <code>\p{GrExt=Y}</code> , <code>\p{GrExt}</code>) (1198)
<code>\p{Greek}</code>	<code>\p{Script=Greek}</code> (Short: <code>\p{Grek}</code> ; NOT <code>\p{Greek_And_Coptic}</code>) (511)
X <code>\p{Greek_And_Coptic}</code>	<code>\p{Block=Greek_And_Coptic}</code> (Short: <code>\p{InGreek}</code>) (144)
X <code>\p{Greek_Extended}</code>	<code>\p{Block=Greek_Extended}</code> (256)

<code>\p{Grek}</code>	<code>\p{Greek}</code> (= <code>\p{Script=Greek}</code>) (NOT <code>\p{Greek_And_Coptic}</code>) (511)
<code>\p{Gujarati}</code>	<code>\p{Script=Gujarati}</code> (Short: <code>\p{Gujr}</code> ; NOT <code>\p{Block=Gujarati}</code>) (83)
<code>\p{Gujr}</code>	<code>\p{Gujarati}</code> (= <code>\p{Script=Gujarati}</code>) (NOT <code>\p{Block=Gujarati}</code>) (83)
<code>\p{Gurmukhi}</code>	<code>\p{Script=Gurmukhi}</code> (Short: <code>\p{Guru}</code> ; NOT <code>\p{Block=Gurmukhi}</code>) (79)
<code>\p{Guru}</code>	<code>\p{Gurmukhi}</code> (= <code>\p{Script=Gurmukhi}</code>) (NOT <code>\p{Block=Gurmukhi}</code>) (79)
X <code>\p{Halfwidth_And_Fullwidth_Forms}</code>	<code>\p{Block=Halfwidth_And_Fullwidth_Forms}</code> (240)
<code>\p{Han}</code>	<code>\p{Script=Han}</code> (75_738)
<code>\p{Hang}</code>	<code>\p{Hangul}</code> (= <code>\p{Script=Hangul}</code>) (11_737)
<code>\p{Hangul}</code>	<code>\p{Script=Hangul}</code> (Short: <code>\p{Hang}</code>) (11_737)
X <code>\p{Hangul_Compatibility_Jamo}</code>	<code>\p{Block=Hangul_Compatibility_Jamo}</code> (96)
X <code>\p{Hangul_Jamo}</code>	<code>\p{Block=Hangul_Jamo}</code> (256)
X <code>\p{Hangul_Jamo_Extended_A}</code>	<code>\p{Block=Hangul_Jamo_Extended_A}</code> (32)
X <code>\p{Hangul_Jamo_Extended_B}</code>	<code>\p{Block=Hangul_Jamo_Extended_B}</code> (80)
<code>\p{Hangul_Syllable_Type: L}</code>	<code>\p{Hangul_Syllable_Type=Leading_Jamo}</code> (125)
<code>\p{Hangul_Syllable_Type: Leading_Jamo}</code>	(Short: <code>\p{Hst=L}</code>) (125)
<code>\p{Hangul_Syllable_Type: LV}</code>	<code>\p{Hangul_Syllable_Type=LV_Syllable}</code> (399)
<code>\p{Hangul_Syllable_Type: LV_Syllable}</code>	(Short: <code>\p{Hst=LV}</code>) (399)
<code>\p{Hangul_Syllable_Type: LVT}</code>	<code>\p{Hangul_Syllable_Type=LVT_Syllable}</code> (10_773)
<code>\p{Hangul_Syllable_Type: LVT_Syllable}</code>	(Short: <code>\p{Hst=LVT}</code>) (10_773)
<code>\p{Hangul_Syllable_Type: NA}</code>	<code>\p{Hangul_Syllable_Type=Not_Applicable}</code> (1_102_583)
<code>\p{Hangul_Syllable_Type: Not_Applicable}</code>	(Short: <code>\p{Hst=NA}</code>) (1_102_583)
<code>\p{Hangul_Syllable_Type: T}</code>	<code>\p{Hangul_Syllable_Type=Trailing_Jamo}</code> (137)
<code>\p{Hangul_Syllable_Type: Trailing_Jamo}</code>	(Short: <code>\p{Hst=T}</code>) (137)
<code>\p{Hangul_Syllable_Type: V}</code>	<code>\p{Hangul_Syllable_Type=Vowel_Jamo}</code> (95)
<code>\p{Hangul_Syllable_Type: Vowel_Jamo}</code>	(Short: <code>\p{Hst=V}</code>) (95)
X <code>\p{Hangul_Syllables}</code>	<code>\p{Block=Hangul_Syllables}</code> (11_184)
<code>\p{Hani}</code>	<code>\p{Han}</code> (= <code>\p{Script=Han}</code>) (75_738)
<code>\p{Hano}</code>	<code>\p{Hanunoo}</code> (= <code>\p{Script=Hanunoo}</code>) (NOT <code>\p{Block=Hanunoo}</code>) (21)
<code>\p{Hanunoo}</code>	<code>\p{Script=Hanunoo}</code> (Short: <code>\p{Hano}</code> ; NOT <code>\p{Block=Hanunoo}</code>) (21)
<code>\p{Hebr}</code>	<code>\p{Hebrew}</code> (= <code>\p{Script=Hebrew}</code>) (NOT <code>\p{Block=Hebrew}</code>) (133)
<code>\p{Hebrew}</code>	<code>\p{Script=Hebrew}</code> (Short: <code>\p{Hebr}</code> ; NOT <code>\p{Block=Hebrew}</code>) (133)
<code>\p{Hex}</code>	<code>\p{XDigit}</code> (= <code>\p{Hex_Digit=Y}</code>) (44)
<code>\p{Hex: *}</code>	<code>\p{Hex_Digit: *}</code>
<code>\p{Hex_Digit}</code>	<code>\p{XDigit}</code> (= <code>\p{Hex_Digit=Y}</code>) (44)
<code>\p{Hex_Digit: N*}</code>	(Short: <code>\p{Hex=N}</code> , <code>\p{Hex}</code>) (1_114_068)
<code>\p{Hex_Digit: Y*}</code>	(Short: <code>\p{Hex=Y}</code> , <code>\p{Hex}</code>) (44)

X	<code>\p{High_Private_Use_Surrogates}</code>	<code>\p{Block=High_Private_Use_Surrogates}</code> (128)
X	<code>\p{High_Surrogates}</code>	<code>\p{Block=High_Surrogates}</code> (896)
	<code>\p{Hira}</code>	<code>\p{Hiragana}</code> (= <code>\p{Script=Hiragana}</code>) (NOT <code>\p{Block=Hiragana}</code>) (90)
	<code>\p{Hiragana}</code>	<code>\p{Script=Hiragana}</code> (Short: <code>\p{Hira}</code> ; NOT <code>\p{Block=Hiragana}</code>) (90)
	<code>\p{HorizSpace}</code>	<code>\p{Blank}</code> (19)
	<code>\p{Hst: *}</code>	<code>\p{Hangul_Syllable_Type: *}</code>
S	<code>\p{Hyphen}</code>	<code>\p{Hyphen=Y}</code> (11)
S	<code>\p{Hyphen: N*}</code>	Use the <code>Line_Break</code> property instead; see www.unicode.org/reports/tr14 (Single: <code>\P{Hyphen}</code>) (1_114_101)
S	<code>\p{Hyphen: Y*}</code>	Use the <code>Line_Break</code> property instead; see www.unicode.org/reports/tr14 (Single: <code>\p{Hyphen}</code>) (11)
	<code>\p{ID_Continue}</code>	<code>\p{ID_Continue=Y}</code> (Short: <code>\p{IDC}</code>) (101_634)
	<code>\p{ID_Continue: N*}</code>	(Short: <code>\p{IDC=N}</code> , <code>\P{IDC}</code>) (1_012_478)
	<code>\p{ID_Continue: Y*}</code>	(Short: <code>\p{IDC=Y}</code> , <code>\p{IDC}</code>) (101_634)
	<code>\p{ID_Start}</code>	<code>\p{ID_Start=Y}</code> (Short: <code>\p{IDS}</code>) (99_764)
	<code>\p{ID_Start: N*}</code>	(Short: <code>\p{IDS=N}</code> , <code>\P{IDS}</code>) (1_014_348)
	<code>\p{ID_Start: Y*}</code>	(Short: <code>\p{IDS=Y}</code> , <code>\p{IDS}</code>) (99_764)
	<code>\p{IDC}</code>	<code>\p{ID_Continue}</code> (= <code>\p{ID_Continue=Y}</code>) (101_634)
	<code>\p{IDC: *}</code>	<code>\p{ID_Continue: *}</code>
	<code>\p{Ideo}</code>	<code>\p{Ideographic}</code> (= <code>\p{Ideographic=Y}</code>) (75_408)
	<code>\p{Ideo: *}</code>	<code>\p{Ideographic: *}</code>
	<code>\p{Ideographic}</code>	<code>\p{Ideographic=Y}</code> (Short: <code>\p{Ideo}</code>) (75_408)
	<code>\p{Ideographic: N*}</code>	(Short: <code>\p{Ideo=N}</code> , <code>\P{Ideo}</code>) (1_038_704)
	<code>\p{Ideographic: Y*}</code>	(Short: <code>\p{Ideo=Y}</code> , <code>\p{Ideo}</code>) (75_408)
X	<code>\p{Ideographic_Description_Characters}</code>	<code>\p{Block=Ideographic_Description_Characters}</code> (16)
	<code>\p{IDS}</code>	<code>\p{ID_Start}</code> (= <code>\p{ID_Start=Y}</code>) (99_764)
	<code>\p{IDS: *}</code>	<code>\p{ID_Start: *}</code>
	<code>\p{IDS_Binary_Operator}</code>	<code>\p{IDS_Binary_Operator=Y}</code> (Short: <code>\p{IDSB}</code>) (10)
	<code>\p{IDS_Binary_Operator: N*}</code>	(Short: <code>\p{IDSB=N}</code> , <code>\P{IDSB}</code>) (1_114_102)
	<code>\p{IDS_Binary_Operator: Y*}</code>	(Short: <code>\p{IDSB=Y}</code> , <code>\p{IDSB}</code>) (10)
	<code>\p{IDS_Tertiary_Operator}</code>	<code>\p{IDS_Tertiary_Operator=Y}</code> (Short: <code>\p{IDST}</code>) (2)
	<code>\p{IDS_Tertiary_Operator: N*}</code>	(Short: <code>\p{IDST=N}</code> , <code>\P{IDST}</code>) (1_114_110)
	<code>\p{IDS_Tertiary_Operator: Y*}</code>	(Short: <code>\p{IDST=Y}</code> , <code>\p{IDST}</code>) (2)
	<code>\p{IDSB}</code>	<code>\p{IDS_Binary_Operator}</code> (= <code>\p{IDS_Binary_Operator=Y}</code>) (10)
	<code>\p{IDSB: *}</code>	<code>\p{IDS_Binary_Operator: *}</code>
	<code>\p{IDST}</code>	<code>\p{IDS_Tertiary_Operator}</code> (= <code>\p{IDS_Tertiary_Operator=Y}</code>) (2)
	<code>\p{IDST: *}</code>	<code>\p{IDS_Tertiary_Operator: *}</code>
	<code>\p{Imperial_Aramaic}</code>	<code>\p{Script=Imperial_Aramaic}</code> (Short: <code>\p{Armi}</code> ; NOT <code>\p{Block=Imperial_Aramaic}</code>) (31)

<code>\p{In: *}</code>	<code>\p{Present_In: *}</code> (Perl extension)
<code>\p{In_*}</code>	<code>\p{Block: *}</code>
<code>\p{Inherited}</code>	<code>\p{Script=Inherited}</code> (Short: <code>\p{Zinh}</code>) (523)
<code>\p{Initial_Punctuation}</code>	<code>\p{General_Category=Initial_Punctuation}</code> (Short: <code>\p{Pi}</code>) (12)
<code>\p{Inscriptional_Pahlavi}</code>	<code>\p{Script=Inscriptional_Pahlavi}</code> (Short: <code>\p{Phli}</code> ; NOT <code>\p{Block=Inscriptional_Pahlavi}</code>) (27)
<code>\p{Inscriptional_Parthian}</code>	<code>\p{Script=Inscriptional_Parthian}</code> (Short: <code>\p{Prti}</code> ; NOT <code>\p{Block=Inscriptional_Parthian}</code>) (30)
X <code>\p{IPA_Extensions}</code>	<code>\p{Block=IPA_Extensions}</code> (96)
<code>\p{Is_*}</code>	<code>\p{*}</code> (Any exceptions are individually noted beginning with the word NOT.) If an entry has flag(s) at its beginning, like 'D', the 'Is_' form has the same flag(s)
<code>\p{Ital}</code>	<code>\p{Old_Italic}</code> (= <code>\p{Script=Old_Italic}</code>) (NOT <code>\p{Block=Old_Italic}</code>) (35)
<code>\p{Java}</code>	<code>\p{Javanese}</code> (= <code>\p{Script=Javanese}</code>) (NOT <code>\p{Block=Javanese}</code>) (91)
<code>\p{Javanese}</code>	<code>\p{Script=Javanese}</code> (Short: <code>\p{Java}</code> ; NOT <code>\p{Block=Javanese}</code>) (91)
<code>\p{Jg: *}</code>	<code>\p{Joining_Group: *}</code>
<code>\p{Join_C}</code>	<code>\p{Join_Control}</code> (= <code>\p{Join_Control=Y}</code>) (2)
<code>\p{Join_C: *}</code>	<code>\p{Join_Control: *}</code>
<code>\p{Join_Control}</code>	<code>\p{Join_Control=Y}</code> (Short: <code>\p{JoinC}</code>) (2)
<code>\p{Join_Control: N*}</code>	(Short: <code>\p{JoinC=N}</code> , <code>\p{JoinC}</code>) (1_114_110)
<code>\p{Join_Control: Y*}</code>	(Short: <code>\p{JoinC=Y}</code> , <code>\p{JoinC}</code>) (2)
<code>\p{Joining_Group: Ain}</code>	(Short: <code>\p{Jg=Ain}</code>) (7)
<code>\p{Joining_Group: Alaph}</code>	(Short: <code>\p{Jg=Alaph}</code>) (1)
<code>\p{Joining_Group: Alef}</code>	(Short: <code>\p{Jg=Alef}</code>) (10)
<code>\p{Joining_Group: Beh}</code>	(Short: <code>\p{Jg=Beh}</code>) (19)
<code>\p{Joining_Group: Beth}</code>	(Short: <code>\p{Jg=Beth}</code>) (2)
<code>\p{Joining_Group: Burushaski_Yeh_Barree}</code>	(Short: <code>\p{Jg=BurushaskiYehBarree}</code>) (2)
<code>\p{Joining_Group: Dal}</code>	(Short: <code>\p{Jg=Dal}</code>) (14)
<code>\p{Joining_Group: Dalath_Rish}</code>	(Short: <code>\p{Jg=DalathRish}</code>) (4)
<code>\p{Joining_Group: E}</code>	(Short: <code>\p{Jg=E}</code>) (1)
<code>\p{Joining_Group: Farsi_Yeh}</code>	(Short: <code>\p{Jg=FarsiYeh}</code>) (7)
<code>\p{Joining_Group: Fe}</code>	(Short: <code>\p{Jg=Fe}</code>) (1)
<code>\p{Joining_Group: Feh}</code>	(Short: <code>\p{Jg=Feh}</code>) (9)
<code>\p{Joining_Group: Final_Semkath}</code>	(Short: <code>\p{Jg=FinalSemkath}</code>) (1)
<code>\p{Joining_Group: Gaf}</code>	(Short: <code>\p{Jg=Gaf}</code>) (13)
<code>\p{Joining_Group: Gamal}</code>	(Short: <code>\p{Jg=Gamal}</code>) (3)
<code>\p{Joining_Group: Hah}</code>	(Short: <code>\p{Jg=Hah}</code>) (17)
<code>\p{Joining_Group: Hamza_On_Heh_Goal}</code>	(Short: <code>\p{Jg=HamzaOnHehGoal}</code>) (1)
<code>\p{Joining_Group: He}</code>	(Short: <code>\p{Jg=He}</code>) (1)
<code>\p{Joining_Group: Heh}</code>	(Short: <code>\p{Jg=Heh}</code>) (1)
<code>\p{Joining_Group: Heh_Goal}</code>	(Short: <code>\p{Jg=HehGoal}</code>) (2)
<code>\p{Joining_Group: Heth}</code>	(Short: <code>\p{Jg=Heth}</code>) (1)
<code>\p{Joining_Group: Kaf}</code>	(Short: <code>\p{Jg=Kaf}</code>) (5)
<code>\p{Joining_Group: Kaph}</code>	(Short: <code>\p{Jg=Kaph}</code>) (1)
<code>\p{Joining_Group: Khaph}</code>	(Short: <code>\p{Jg=Khaph}</code>) (1)

```

\p{Joining_Group: Knotted_Heh} (Short: \p{Jg=KnottedHeh}) (2)
\p{Joining_Group: Lam} (Short: \p{Jg=Lam}) (6)
\p{Joining_Group: Lamadh} (Short: \p{Jg=Lamadh}) (1)
\p{Joining_Group: Meem} (Short: \p{Jg=Meem}) (3)
\p{Joining_Group: Mim} (Short: \p{Jg=Mim}) (1)
\p{Joining_Group: No_Joining_Group} (Short: \p{Jg=NoJoiningGroup})
(1_113_883)
\p{Joining_Group: Noon} (Short: \p{Jg=Noon}) (8)
\p{Joining_Group: Nun} (Short: \p{Jg=Nun}) (1)
\p{Joining_Group: Nya} (Short: \p{Jg=Nya}) (1)
\p{Joining_Group: Pe} (Short: \p{Jg=Pe}) (1)
\p{Joining_Group: Qaf} (Short: \p{Jg=Qaf}) (4)
\p{Joining_Group: Qaph} (Short: \p{Jg=Qaph}) (1)
\p{Joining_Group: Reh} (Short: \p{Jg=Reh}) (16)
\p{Joining_Group: Reversed_Pe} (Short: \p{Jg=ReversedPe}) (1)
\p{Joining_Group: Sad} (Short: \p{Jg=Sad}) (5)
\p{Joining_Group: Sadhe} (Short: \p{Jg=Sadhe}) (1)
\p{Joining_Group: Seen} (Short: \p{Jg=Seen}) (11)
\p{Joining_Group: Semkath} (Short: \p{Jg=Semkath}) (1)
\p{Joining_Group: Shin} (Short: \p{Jg=Shin}) (1)
\p{Joining_Group: Swash_Kaf} (Short: \p{Jg=SwashKaf}) (1)
\p{Joining_Group: Syriac_Waw} (Short: \p{Jg=SyriacWaw}) (1)
\p{Joining_Group: Tah} (Short: \p{Jg=Tah}) (3)
\p{Joining_Group: Taw} (Short: \p{Jg=Taw}) (1)
\p{Joining_Group: Teh_Marbuta} (Short: \p{Jg=TehMarbuta}) (3)
\p{Joining_Group: Teth} (Short: \p{Jg=Teth}) (2)
\p{Joining_Group: Waw} (Short: \p{Jg=Waw}) (15)
\p{Joining_Group: Yeh} (Short: \p{Jg=Yeh}) (7)
\p{Joining_Group: Yeh_Barree} (Short: \p{Jg=YehBarree}) (2)
\p{Joining_Group: Yeh_With_Tail} (Short: \p{Jg=YehWithTail}) (1)
\p{Joining_Group: Yudh} (Short: \p{Jg=Yudh}) (1)
\p{Joining_Group: Yudh_He} (Short: \p{Jg=YudhHe}) (1)
\p{Joining_Group: Zain} (Short: \p{Jg=Zain}) (1)
\p{Joining_Group: Zhain} (Short: \p{Jg=Zhain}) (1)
\p{Joining_Type: C} \p{Joining_Type=Join_Causing} (3)
\p{Joining_Type: D} \p{Joining_Type=Dual_Joining} (188)
\p{Joining_Type: Dual_Joining} (Short: \p{Jt=D}) (188)
\p{Joining_Type: Join_Causing} (Short: \p{Jt=C}) (3)
\p{Joining_Type: L} \p{Joining_Type=Left_Joining} (0)
\p{Joining_Type: Left_Joining} (Short: \p{Jt=L}) (0)
\p{Joining_Type: Non_Joining} (Short: \p{Jt=U}) (1_112_539)
\p{Joining_Type: R} \p{Joining_Type=Right_Joining} (74)
\p{Joining_Type: Right_Joining} (Short: \p{Jt=R}) (74)
\p{Joining_Type: T} \p{Joining_Type=Transparent} (1308)
\p{Joining_Type: Transparent} (Short: \p{Jt=T}) (1308)
\p{Joining_Type: U} \p{Joining_Type=Non_Joining} (1_112_539)
\p{Jt: *} \p{Joining_Type: *}
\p{Kaithi} \p{Script=Kaithi} (Short: \p{Kthi}; NOT
\p{Block=Kaithi}) (66)
\p{Kali} \p{Kayah_Li} (= \p{Script=Kayah_Li}) (48)
\p{Kana} \p{Katakana} (= \p{Script=Katakana}) (NOT
\p{Block=Katakana}) (299)
X \p{Kanbun} \p{Block=Kanbun} (16)
X \p{Kangxi_Radicals} \p{Block=Kangxi_Radicals} (224)
\p{Kannada} \p{Script=Kannada} (Short: \p{Knda}; NOT
\p{Block=Kannada}) (84)

```

	<code>\p{Katakana}</code>	<code>\p{Script=Katakana}</code> (Short: <code>\p{Kana}</code> ; NOT <code>\p{Block=Katakana}</code>) (299)
X	<code>\p{Katakana_Phonetic_Extensions}</code>	<code>\p{Block=Katakana_Phonetic_Extensions}</code> (16)
	<code>\p{Kayah_Li}</code>	<code>\p{Script=Kayah_Li}</code> (Short: <code>\p{Kali}</code>) (48)
	<code>\p{Khar}</code>	<code>\p{Kharoshthi}</code> (= <code>\p{Script=Kharoshthi}</code>) (NOT <code>\p{Block=Kharoshthi}</code>) (65)
	<code>\p{Kharoshthi}</code>	<code>\p{Script=Kharoshthi}</code> (Short: <code>\p{Khar}</code> ; NOT <code>\p{Block=Kharoshthi}</code>) (65)
	<code>\p{Khmer}</code>	<code>\p{Script=Khmer}</code> (Short: <code>\p{Khmr}</code> ; NOT <code>\p{Block=Khmer}</code>) (146)
X	<code>\p{Khmer_Symbols}</code>	<code>\p{Block=Khmer_Symbols}</code> (32)
	<code>\p{Khmr}</code>	<code>\p{Khmer}</code> (= <code>\p{Script=Khmer}</code>) (NOT <code>\p{Block=Khmer}</code>) (146)
	<code>\p{Knda}</code>	<code>\p{Kannada}</code> (= <code>\p{Script=Kannada}</code>) (NOT <code>\p{Block=Kannada}</code>) (84)
	<code>\p{Kthi}</code>	<code>\p{Kaithi}</code> (= <code>\p{Script=Kaithi}</code>) (NOT <code>\p{Block=Kaithi}</code>) (66)
	<code>\p{L}</code>	<code>\p{Letter}</code> (= <code>\p{General_Category=Letter}</code>) (99_537)
	<code>\p{L&}</code>	<code>\p{Cased_Letter}</code> (= <code>\p{General_Category=Cased_Letter}</code>) (3207)
	<code>\p{L_}</code>	<code>\p{Cased_Letter}</code> (= <code>\p{General_Category=Cased_Letter}</code>) (3207)
	<code>\p{Lana}</code>	<code>\p{Tai_Tham}</code> (= <code>\p{Script=Tai_Tham}</code>) (NOT <code>\p{Block=Tai_Tham}</code>) (127)
	<code>\p{Lao}</code>	<code>\p{Script=Lao}</code> (NOT <code>\p{Block=Lao}</code>) (65)
	<code>\p{Laoo}</code>	<code>\p{Lao}</code> (= <code>\p{Script=Lao}</code>) (NOT <code>\p{Block=Lao}</code>) (65)
	<code>\p{Latin}</code>	<code>\p{Script=Latin}</code> (Short: <code>\p{Latn}</code>) (1244)
X	<code>\p{Latin_1}</code>	<code>\p{Latin_1_Supplement}</code> (= <code>\p{Block=Latin_1_Supplement}</code>) (128)
X	<code>\p{Latin_1_Supplement}</code>	<code>\p{Block=Latin_1_Supplement}</code> (Short: <code>\p{InLatin1}</code>) (128)
X	<code>\p{Latin_Extended_A}</code>	<code>\p{Block=Latin_Extended_A}</code> (128)
X	<code>\p{Latin_Extended_Additional}</code>	<code>\p{Block=Latin_Extended_Additional}</code> (256)
X	<code>\p{Latin_Extended_B}</code>	<code>\p{Block=Latin_Extended_B}</code> (208)
X	<code>\p{Latin_Extended_C}</code>	<code>\p{Block=Latin_Extended_C}</code> (32)
X	<code>\p{Latin_Extended_D}</code>	<code>\p{Block=Latin_Extended_D}</code> (224)
	<code>\p{Latn}</code>	<code>\p{Latin}</code> (= <code>\p{Script=Latin}</code>) (1244)
	<code>\p{Lb: *}</code>	<code>\p{Line_Break: *}</code>
	<code>\p{LC}</code>	<code>\p{Cased_Letter}</code> (= <code>\p{General_Category=Cased_Letter}</code>) (3207)
	<code>\p{Lepc}</code>	<code>\p{Lepcha}</code> (= <code>\p{Script=Lepcha}</code>) (NOT <code>\p{Block=Lepcha}</code>) (74)
	<code>\p{Lepcha}</code>	<code>\p{Script=Lepcha}</code> (Short: <code>\p{Lepc}</code> ; NOT <code>\p{Block=Lepcha}</code>) (74)
	<code>\p{Letter}</code>	<code>\p{General_Category=Letter}</code> (Short: <code>\p{L}</code>) (99_537)
	<code>\p{Letter_Number}</code>	<code>\p{General_Category=Letter_Number}</code> (Short: <code>\p{NL}</code>) (224)
X	<code>\p{Letterlike_Symbols}</code>	<code>\p{Block=Letterlike_Symbols}</code> (80)
	<code>\p{Limb}</code>	<code>\p{Limbu}</code> (= <code>\p{Script=Limbu}</code>) (NOT <code>\p{Block=Limbu}</code>) (66)
	<code>\p{Limbu}</code>	<code>\p{Script=Limbu}</code> (Short: <code>\p{Limb}</code> ; NOT

`\p{Block=Limbu}` (66)
`\p{Linb}`
`\p{Line_Break: AI}` `\p{Line_Break=Linear_B}` (= `\p{Script=Linear_B}`) (211)
`\p{Line_Break: AL}` `\p{Line_Break=Ambiguous}` (644)
`\p{Line_Break: Alphabetic}` (Short: `\p{Lb=AL}`) (14_092)
`\p{Line_Break: Ambiguous}` (Short: `\p{Lb=AI}`) (644)
`\p{Line_Break: B2}` `\p{Line_Break=Break_Both}` (1)
`\p{Line_Break: BA}` `\p{Line_Break=Break_After}` (137)
`\p{Line_Break: BB}` `\p{Line_Break=Break_Before}` (19)
`\p{Line_Break: BK}` `\p{Line_Break=Mandatory_Break}` (4)
`\p{Line_Break: Break_After}` (Short: `\p{Lb=BA}`) (137)
`\p{Line_Break: Break_Before}` (Short: `\p{Lb=BB}`) (19)
`\p{Line_Break: Break_Both}` (Short: `\p{Lb=B2}`) (1)
`\p{Line_Break: Break_Symbols}` (Short: `\p{Lb=SY}`) (1)
`\p{Line_Break: Carriage_Return}` (Short: `\p{Lb=CR}`) (1)
`\p{Line_Break: CB}` `\p{Line_Break=Contingent_Break}` (1)
`\p{Line_Break: CL}` `\p{Line_Break=Close_Punctuation}` (87)
`\p{Line_Break: Close_Parenthesis}` (Short: `\p{Lb=CP}`) (2)
`\p{Line_Break: Close_Punctuation}` (Short: `\p{Lb=CL}`) (87)
`\p{Line_Break: CM}` `\p{Line_Break=Combining_Mark}` (1436)
`\p{Line_Break: Combining_Mark}` (Short: `\p{Lb=CM}`) (1436)
`\p{Line_Break: Complex_Context}` (Short: `\p{Lb=SA}`) (662)
`\p{Line_Break: Contingent_Break}` (Short: `\p{Lb=CB}`) (1)
`\p{Line_Break: CP}` `\p{Line_Break=Close_Parenthesis}` (2)
`\p{Line_Break: CR}` `\p{Line_Break=Carriage_Return}` (1)
`\p{Line_Break: EX}` `\p{Line_Break=Exclamation}` (34)
`\p{Line_Break: Exclamation}` (Short: `\p{Lb=EX}`) (34)
`\p{Line_Break: GL}` `\p{Line_Break=Glue}` (16)
`\p{Line_Break: Glue}` (Short: `\p{Lb=GL}`) (16)
`\p{Line_Break: H2}` (Short: `\p{Lb=H2}`) (399)
`\p{Line_Break: H3}` (Short: `\p{Lb=H3}`) (10_773)
`\p{Line_Break: HY}` `\p{Line_Break=Hyphen}` (1)
`\p{Line_Break: Hyphen}` (Short: `\p{Lb=HY}`) (1)
`\p{Line_Break: ID}` `\p{Line_Break=Ideographic}` (161_775)
`\p{Line_Break: Ideographic}` (Short: `\p{Lb=ID}`) (161_775)
`\p{Line_Break: IN}` `\p{Line_Break=Inseparable}` (4)
`\p{Line_Break: Infix_Numeric}` (Short: `\p{Lb=IS}`) (13)
`\p{Line_Break: Inseparable}` (Short: `\p{Lb=IN}`) (4)
`\p{Line_Break: Inseparable}` `\p{Line_Break=Inseparable}` (4)
`\p{Line_Break: IS}` `\p{Line_Break=Infix_Numeric}` (13)
`\p{Line_Break: JL}` (Short: `\p{Lb=JL}`) (125)
`\p{Line_Break: JT}` (Short: `\p{Lb=JT}`) (137)
`\p{Line_Break: JV}` (Short: `\p{Lb=JV}`) (95)
`\p{Line_Break: LF}` `\p{Line_Break=Line_Feed}` (1)
`\p{Line_Break: Line_Feed}` (Short: `\p{Lb=LF}`) (1)
`\p{Line_Break: Mandatory_Break}` (Short: `\p{Lb=BK}`) (4)
`\p{Line_Break: Next_Line}` (Short: `\p{Lb=NL}`) (1)
`\p{Line_Break: NL}` `\p{Line_Break=Next_Line}` (1)
`\p{Line_Break: Nonstarter}` (Short: `\p{Lb=NS}`) (77)
`\p{Line_Break: NS}` `\p{Line_Break=Nonstarter}` (77)
`\p{Line_Break: NU}` `\p{Line_Break=Numeric}` (403)
`\p{Line_Break: Numeric}` (Short: `\p{Lb=NU}`) (403)
`\p{Line_Break: OP}` `\p{Line_Break=Open_Punctuation}` (81)
`\p{Line_Break: Open_Punctuation}` (Short: `\p{Lb=OP}`) (81)
`\p{Line_Break: PO}` `\p{Line_Break=Postfix_Numeric}` (28)
`\p{Line_Break: Postfix_Numeric}` (Short: `\p{Lb=PO}`) (28)

	<code>\p{Line_Break: PR}</code>	<code>\p{Line_Break=Prefix_Numeric}</code> (43)
	<code>\p{Line_Break: Prefix_Numeric}</code>	(Short: <code>\p{Lb=PR}</code>) (43)
	<code>\p{Line_Break: QU}</code>	<code>\p{Line_Break=Quotation}</code> (34)
	<code>\p{Line_Break: Quotation}</code>	(Short: <code>\p{Lb=QU}</code>) (34)
	<code>\p{Line_Break: SA}</code>	<code>\p{Line_Break=Complex_Context}</code> (662)
D	<code>\p{Line_Break: SG}</code>	<code>\p{Line_Break=Surrogate}</code> (2048)
	<code>\p{Line_Break: SP}</code>	<code>\p{Line_Break=Space}</code> (1)
	<code>\p{Line_Break: Space}</code>	(Short: <code>\p{Lb=SP}</code>) (1)
D	<code>\p{Line_Break: Surrogate}</code>	Deprecated by Unicode because surrogates should never appear in well-formed text, and therefore shouldn't be the basis for line breaking (Short: <code>\p{Lb=SG}</code>) (2048)
	<code>\p{Line_Break: SY}</code>	<code>\p{Line_Break=Break_Symbols}</code> (1)
	<code>\p{Line_Break: Unknown}</code>	(Short: <code>\p{Lb=XX}</code>) (920_933)
	<code>\p{Line_Break: WJ}</code>	<code>\p{Line_Break=Word_Joiner}</code> (2)
	<code>\p{Line_Break: Word_Joiner}</code>	(Short: <code>\p{Lb=WJ}</code>) (2)
	<code>\p{Line_Break: XX}</code>	<code>\p{Line_Break=Unknown}</code> (920_933)
	<code>\p{Line_Break: ZW}</code>	<code>\p{Line_Break=ZWSpace}</code> (1)
	<code>\p{Line_Break: ZWSpace}</code>	(Short: <code>\p{Lb=ZW}</code>) (1)
	<code>\p{Line_Separator}</code>	<code>\p{General_Category=Line_Separator}</code> (Short: <code>\p{Zl}</code>) (1)
	<code>\p{Linear_B}</code>	<code>\p{Script=Linear_B}</code> (Short: <code>\p{Linb}</code>) (211)
X	<code>\p{Linear_B_Ideograms}</code>	<code>\p{Block=Linear_B_Ideograms}</code> (128)
X	<code>\p{Linear_B_Syllabary}</code>	<code>\p{Block=Linear_B_Syllabary}</code> (128)
	<code>\p{Lisu}</code>	<code>\p{Script=Lisu}</code> (48)
	<code>\p{Ll}</code>	<code>\p{Lowercase_Letter}</code> (= <code>\p{General_Category=Lowercase_Letter}</code>) (1749)
	<code>\p{Lm}</code>	<code>\p{Modifier_Letter}</code> (= <code>\p{General_Category=Modifier_Letter}</code>) (202)
	<code>\p{Lo}</code>	<code>\p{Other_Letter}</code> (= <code>\p{General_Category=Other_Letter}</code>) (96_128)
	<code>\p{LOE}</code>	<code>\p{Logical_Order_Exception}</code> (= <code>\p{Logical_Order_Exception=Y}</code>) (15)
	<code>\p{LOE: *}</code>	<code>\p{Logical_Order_Exception: *}</code>
	<code>\p{Logical_Order_Exception}</code>	<code>\p{Logical_Order_Exception=Y}</code> (Short: <code>\p{LOE}</code>) (15)
	<code>\p{Logical_Order_Exception: N*}</code>	(Short: <code>\p{LOE=N}</code> , <code>\p{LOE}</code>) (1_114_097)
	<code>\p{Logical_Order_Exception: Y*}</code>	(Short: <code>\p{LOE=Y}</code> , <code>\p{LOE}</code>) (15)
X	<code>\p{Low_Surrogates}</code>	<code>\p{Block=Low_Surrogates}</code> (1024)
	<code>\p{Lower}</code>	<code>\p{Lowercase=Y}</code> (1908)
	<code>\p{Lower: *}</code>	<code>\p{Lowercase: *}</code>
	<code>\p{Lowercase}</code>	<code>\p{Lower}</code> (= <code>\p{Lowercase=Y}</code>) (1908)
	<code>\p{Lowercase: N*}</code>	(Short: <code>\p{Lower=N}</code> , <code>\p{Lower}</code>) (1_112_204)
	<code>\p{Lowercase: Y*}</code>	(Short: <code>\p{Lower=Y}</code> , <code>\p{Lower}</code>) (1908)
	<code>\p{Lowercase_Letter}</code>	<code>\p{General_Category=Lowercase_Letter}</code> (Short: <code>\p{Ll}</code>) (1749)
	<code>\p{Lt}</code>	<code>\p{Title}</code> (= <code>\p{General_Category=Titlecase_Letter}</code>) (31)
	<code>\p{Lu}</code>	<code>\p{Uppercase_Letter}</code> (= <code>\p{General_Category=Uppercase_Letter}</code>) (1427)
	<code>\p{Lyci}</code>	<code>\p{Lycian}</code> (= <code>\p{Script=Lycian}</code>) (NOT <code>\p{Block=Lycian}</code>) (29)

<code>\p{Lycian}</code>	<code>\p{Script=Lycian}</code> (Short: <code>\p{Lyci}</code> ; NOT <code>\p{Block=Lycian}</code>) (29)
<code>\p{Lydi}</code>	<code>\p{Lydian}</code> (= <code>\p{Script=Lydian}</code>) (NOT <code>\p{Block=Lydian}</code>) (27)
<code>\p{Lydian}</code>	<code>\p{Script=Lydian}</code> (Short: <code>\p{Lydi}</code> ; NOT <code>\p{Block=Lydian}</code>) (27)
<code>\p{M}</code>	<code>\p{Mark}</code> (= <code>\p{General_Category=Mark}</code>) (1451)
X <code>\p{Mahjong_Tiles}</code>	<code>\p{Block=Mahjong_Tiles}</code> (48)
<code>\p{Malayalam}</code>	<code>\p{Script=Malayalam}</code> (Short: <code>\p{Mlym}</code> ; NOT <code>\p{Block=Malayalam}</code>) (95)
<code>\p{Mark}</code>	<code>\p{General_Category=Mark}</code> (Short: <code>\p{M}</code>) (1451)
<code>\p{Math}</code>	<code>\p{Math=Y}</code> (2161)
<code>\p{Math: N*}</code>	(Single: <code>\p{Math}</code>) (1_111_951)
<code>\p{Math: Y*}</code>	(Single: <code>\p{Math}</code>) (2161)
<code>\p{Math_Symbol}</code>	<code>\p{General_Category=Math_Symbol}</code> (Short: <code>\p{Sm}</code>) (945)
X <code>\p{Mathematical_Alphanumeric_Symbols}</code>	<code>\p{Block=Mathematical_Alphanumeric_Symbols}</code> (1024)
X <code>\p{Mathematical_Operators}</code>	<code>\p{Block=Mathematical_Operators}</code> (256)
<code>\p{Mc}</code>	<code>\p{Spacing_Mark}</code> (= <code>\p{General_Category=Spacing_Mark}</code>) (276)
<code>\p{Me}</code>	<code>\p{Enclosing_Mark}</code> (= <code>\p{General_Category=Enclosing_Mark}</code>) (13)
<code>\p{Meetei_Mayek}</code>	<code>\p{Script=Meetei_Mayek}</code> (Short: <code>\p{Mtei}</code> ; NOT <code>\p{Block=Meetei_Mayek}</code>) (56)
X <code>\p{Miscellaneous_Mathematical_Symbols_A}</code>	<code>\p{Block=Miscellaneous_Mathematical_Symbols_A}</code> (48)
X <code>\p{Miscellaneous_Mathematical_Symbols_B}</code>	<code>\p{Block=Miscellaneous_Mathematical_Symbols_B}</code> (128)
X <code>\p{Miscellaneous_Symbols}</code>	<code>\p{Block=Miscellaneous_Symbols}</code> (256)
X <code>\p{Miscellaneous_Symbols_And_Arrows}</code>	<code>\p{Block=Miscellaneous_Symbols_And_Arrows}</code> (256)
X <code>\p{Miscellaneous_Technical}</code>	<code>\p{Block=Miscellaneous_Technical}</code> (256)
<code>\p{Mlym}</code>	<code>\p{Malayalam}</code> (= <code>\p{Script=Malayalam}</code>) (NOT <code>\p{Block=Malayalam}</code>) (95)
<code>\p{Mn}</code>	<code>\p{Nonspacing_Mark}</code> (= <code>\p{General_Category=Nonspacing_Mark}</code>) (1162)
<code>\p{Modifier_Letter}</code>	<code>\p{General_Category=Modifier_Letter}</code> (Short: <code>\p{Lm}</code>) (202)
<code>\p{Modifier_Symbol}</code>	<code>\p{General_Category=Modifier_Symbol}</code> (Short: <code>\p{Sk}</code>) (99)
X <code>\p{Modifier_Tone_Letters}</code>	<code>\p{Block=Modifier_Tone_Letters}</code> (32)
<code>\p{Mong}</code>	<code>\p{Mongolian}</code> (= <code>\p{Script=Mongolian}</code>) (NOT <code>\p{Block=Mongolian}</code>) (153)
<code>\p{Mongolian}</code>	<code>\p{Script=Mongolian}</code> (Short: <code>\p{Mong}</code> ; NOT <code>\p{Block=Mongolian}</code>) (153)
<code>\p{Mtei}</code>	<code>\p{Meetei_Mayek}</code> (= <code>\p{Script=Meetei_Mayek}</code>) (NOT <code>\p{Block=Meetei_Mayek}</code>) (56)
X <code>\p{Musical_Symbols}</code>	<code>\p{Block=Musical_Symbols}</code> (256)
<code>\p{Myanmar}</code>	<code>\p{Script=Myanmar}</code> (Short: <code>\p{Mymr}</code> ; NOT

	<code>\p{Block=Myanmar}</code>) (188)
X <code>\p{Myanmar_Extended_A}</code>	<code>\p{Block=Myanmar_Extended_A}</code> (32)
<code>\p{Mymr}</code>	<code>\p{Myanmar}</code> (= <code>\p{Script=Myanmar}</code>) (NOT <code>\p{Block=Myanmar}</code>) (188)
<code>\p{N}</code>	<code>\p{Number}</code> (= <code>\p{General_Category=Number}</code>) (1064)
<code>\p{NChar}</code>	<code>\p{Noncharacter_Code_Point}</code> (= <code>\p{Noncharacter_Code_Point=Y}</code>) (66)
<code>\p{NChar: *}</code>	<code>\p{Noncharacter_Code_Point: *}</code>
<code>\p{Nd}</code>	<code>\p{Digit}</code> (= <code>\p{General_Category=Decimal_Number}</code>) (411)
<code>\p{New_Tai_Lue}</code>	<code>\p{Script=New_Tai_Lue}</code> (Short: <code>\p{Talu}</code> ; NOT <code>\p{Block=New_Tai_Lue}</code>) (83)
<code>\p{NFC_QC: *}</code>	<code>\p{NFC_Quick_Check: *}</code>
<code>\p{NFC_Quick_Check: M}</code>	<code>\p{NFC_Quick_Check=Maybe}</code> (103)
<code>\p{NFC_Quick_Check: Maybe}</code>	(Short: <code>\p{NFCQC=M}</code>) (103)
<code>\p{NFC_Quick_Check: N}</code>	<code>\p{NFC_Quick_Check=No}</code> (NOT <code>\p{NFC_Quick_Check}</code> NOR <code>\p{NFC_QC}</code> NOR <code>\p{Is_NFC_Quick_Check}</code> NOR <code>\p{Is_NFC_QC}</code>) (1118)
<code>\p{NFC_Quick_Check: No}</code>	(Short: <code>\p{NFCQC=N}</code> ; NOT <code>\p{NFC_Quick_Check}</code> NOR <code>\p{NFC_QC}</code> NOR <code>\p{Is_NFC_Quick_Check}</code> NOR <code>\p{Is_NFC_QC}</code>) (1118)
<code>\p{NFC_Quick_Check: Y}</code>	<code>\p{NFC_Quick_Check=Yes}</code> (NOT <code>\p{NFC_Quick_Check}</code> NOR <code>\p{NFC_QC}</code> NOR <code>\p{Is_NFC_Quick_Check}</code> NOR <code>\p{Is_NFC_QC}</code>) (1_112_891)
<code>\p{NFC_Quick_Check: Yes}</code>	(Short: <code>\p{NFCQC=Y}</code> ; NOT <code>\p{NFC_Quick_Check}</code> NOR <code>\p{NFC_QC}</code> NOR <code>\p{Is_NFC_Quick_Check}</code> NOR <code>\p{Is_NFC_QC}</code>) (1_112_891)
<code>\p{NFD_QC: *}</code>	<code>\p{NFD_Quick_Check: *}</code>
<code>\p{NFD_Quick_Check: N}</code>	<code>\p{NFD_Quick_Check=No}</code> (NOT <code>\p{NFD_Quick_Check}</code> NOR <code>\p{NFD_QC}</code> NOR <code>\p{Is_NFD_Quick_Check}</code> NOR <code>\p{Is_NFD_QC}</code>) (13_221)
<code>\p{NFD_Quick_Check: No}</code>	(Short: <code>\p{NFDQC=N}</code> ; NOT <code>\p{NFD_Quick_Check}</code> NOR <code>\p{NFD_QC}</code> NOR <code>\p{Is_NFD_Quick_Check}</code> NOR <code>\p{Is_NFD_QC}</code>) (13_221)
<code>\p{NFD_Quick_Check: Y}</code>	<code>\p{NFD_Quick_Check=Yes}</code> (NOT <code>\p{NFD_Quick_Check}</code> NOR <code>\p{NFD_QC}</code> NOR <code>\p{Is_NFD_Quick_Check}</code> NOR <code>\p{Is_NFD_QC}</code>) (1_100_891)
<code>\p{NFD_Quick_Check: Yes}</code>	(Short: <code>\p{NFDQC=Y}</code> ; NOT <code>\p{NFD_Quick_Check}</code> NOR <code>\p{NFD_QC}</code> NOR <code>\p{Is_NFD_Quick_Check}</code> NOR <code>\p{Is_NFD_QC}</code>) (1_100_891)
<code>\p{NFKC_QC: *}</code>	<code>\p{NFKC_Quick_Check: *}</code>
<code>\p{NFKC_Quick_Check: M}</code>	<code>\p{NFKC_Quick_Check=Maybe}</code> (103)
<code>\p{NFKC_Quick_Check: Maybe}</code>	(Short: <code>\p{NFKCQC=M}</code>) (103)
<code>\p{NFKC_Quick_Check: N}</code>	<code>\p{NFKC_Quick_Check=No}</code> (NOT <code>\p{NFKC_Quick_Check}</code> NOR <code>\p{NFKC_QC}</code> NOR <code>\p{Is_NFKC_Quick_Check}</code> NOR <code>\p{Is_NFKC_QC}</code>) (4597)

<code>\p{NFKC_Quick_Check: No}</code>	<code>(Short: \p{NFKCQC=N}; NOT \P{NFKC_Quick_Check} NOR \P{NFKC_QC} NOR \P{Is_NFKC_Quick_Check} NOR \P{Is_NFKC_QC}) (4597)</code>
<code>\p{NFKC_Quick_Check: Y}</code>	<code>\p{NFKC_Quick_Check=Yes} (NOT \P{NFKC_Quick_Check} NOR \P{NFKC_QC} NOR \P{Is_NFKC_Quick_Check} NOR \P{Is_NFKC_QC}) (1_109_412)</code>
<code>\p{NFKC_Quick_Check: Yes}</code>	<code>(Short: \p{NFKCQC=Y}; NOT \P{NFKC_Quick_Check} NOR \P{NFKC_QC} NOR \P{Is_NFKC_Quick_Check} NOR \P{Is_NFKC_QC}) (1_109_412)</code>
<code>\p{NFKD_QC: *}</code>	<code>\p{NFKD_Quick_Check: *}</code>
<code>\p{NFKD_Quick_Check: N}</code>	<code>\p{NFKD_Quick_Check=No} (NOT \P{NFKD_Quick_Check} NOR \P{NFKD_QC} NOR \P{Is_NFKD_Quick_Check} NOR \P{Is_NFKD_QC}) (16_688)</code>
<code>\p{NFKD_Quick_Check: No}</code>	<code>(Short: \p{NFKDQC=N}; NOT \P{NFKD_Quick_Check} NOR \P{NFKD_QC} NOR \P{Is_NFKD_Quick_Check} NOR \P{Is_NFKD_QC}) (16_688)</code>
<code>\p{NFKD_Quick_Check: Y}</code>	<code>\p{NFKD_Quick_Check=Yes} (NOT \P{NFKD_Quick_Check} NOR \P{NFKD_QC} NOR \P{Is_NFKD_Quick_Check} NOR \P{Is_NFKD_QC}) (1_097_424)</code>
<code>\p{NFKD_Quick_Check: Yes}</code>	<code>(Short: \p{NFKDQC=Y}; NOT \P{NFKD_Quick_Check} NOR \P{NFKD_QC} NOR \P{Is_NFKD_Quick_Check} NOR \P{Is_NFKD_QC}) (1_097_424)</code>
<code>\p{Nko}</code>	<code>\p{Script=Nko} (NOT \p{Nko}) (59)</code>
<code>\p{Nkoo}</code>	<code>\p{Nko} (= \p{Script=Nko}) (NOT \p{Nko}) (59)</code>
<code>\p{Nl}</code>	<code>\p{Letter_Number} (= \p{General_Category= Letter_Number}) (224)</code>
<code>\p{No}</code>	<code>\p{Other_Number} (= \p{General_Category= Other_Number}) (429)</code>
X <code>\p{No_Block}</code>	<code>\p{Block=No_Block} (864_192)</code>
<code>\p{Noncharacter_Code_Point}</code>	<code>\p{Noncharacter_Code_Point=Y} (Short: \p{NChar}) (66)</code>
<code>\p{Noncharacter_Code_Point: N*}</code>	<code>(Short: \p{NChar=N}, \P{NChar}) (1_114_046)</code>
<code>\p{Noncharacter_Code_Point: Y*}</code>	<code>(Short: \p{NChar=Y}, \p{NChar}) (66)</code>
<code>\p{Nonspacing_Mark}</code>	<code>\p{General_Category=Nonspacing_Mark} (Short: \p{Mn}) (1162)</code>
<code>\p{Nt: *}</code>	<code>\p{Numeric_Type: *}</code>
<code>\p{Number}</code>	<code>\p{General_Category=Number} (Short: \p{N}) (1064)</code>
X <code>\p{Number_Forms}</code>	<code>\p{Block=Number_Forms} (64)</code>
<code>\p{Numeric_Type: De}</code>	<code>\p{Numeric_Type=Decimal} (411)</code>
<code>\p{Numeric_Type: Decimal}</code>	<code>(Short: \p{Nt=De}) (411)</code>
<code>\p{Numeric_Type: Di}</code>	<code>\p{Numeric_Type=Digit} (118)</code>
<code>\p{Numeric_Type: Digit}</code>	<code>(Short: \p{Nt=Di}) (118)</code>
<code>\p{Numeric_Type: None}</code>	<code>(Short: \p{Nt=None}) (1_112_971)</code>
<code>\p{Numeric_Type: Nu}</code>	<code>\p{Numeric_Type=Numeric} (612)</code>
<code>\p{Numeric_Type: Numeric}</code>	<code>(Short: \p{Nt=Nu}) (612)</code>

T `\p{Numeric_Value: 27}` (Short: `\p{Nv=27}`) (1)
T `\p{Numeric_Value: 28}` (Short: `\p{Nv=28}`) (1)
T `\p{Numeric_Value: 29}` (Short: `\p{Nv=29}`) (1)
T `\p{Numeric_Value: 30}` (Short: `\p{Nv=30}`) (9)
T `\p{Numeric_Value: 31}` (Short: `\p{Nv=31}`) (1)
T `\p{Numeric_Value: 32}` (Short: `\p{Nv=32}`) (1)
T `\p{Numeric_Value: 33}` (Short: `\p{Nv=33}`) (1)
T `\p{Numeric_Value: 34}` (Short: `\p{Nv=34}`) (1)
T `\p{Numeric_Value: 35}` (Short: `\p{Nv=35}`) (1)
T `\p{Numeric_Value: 36}` (Short: `\p{Nv=36}`) (1)
T `\p{Numeric_Value: 37}` (Short: `\p{Nv=37}`) (1)
T `\p{Numeric_Value: 38}` (Short: `\p{Nv=38}`) (1)
T `\p{Numeric_Value: 39}` (Short: `\p{Nv=39}`) (1)
T `\p{Numeric_Value: 40}` (Short: `\p{Nv=40}`) (8)
T `\p{Numeric_Value: 41}` (Short: `\p{Nv=41}`) (1)
T `\p{Numeric_Value: 42}` (Short: `\p{Nv=42}`) (1)
T `\p{Numeric_Value: 43}` (Short: `\p{Nv=43}`) (1)
T `\p{Numeric_Value: 44}` (Short: `\p{Nv=44}`) (1)
T `\p{Numeric_Value: 45}` (Short: `\p{Nv=45}`) (1)
T `\p{Numeric_Value: 46}` (Short: `\p{Nv=46}`) (1)
T `\p{Numeric_Value: 47}` (Short: `\p{Nv=47}`) (1)
T `\p{Numeric_Value: 48}` (Short: `\p{Nv=48}`) (1)
T `\p{Numeric_Value: 49}` (Short: `\p{Nv=49}`) (1)
T `\p{Numeric_Value: 50}` (Short: `\p{Nv=50}`) (18)
T `\p{Numeric_Value: 60}` (Short: `\p{Nv=60}`) (4)
T `\p{Numeric_Value: 70}` (Short: `\p{Nv=70}`) (4)
T `\p{Numeric_Value: 80}` (Short: `\p{Nv=80}`) (4)
T `\p{Numeric_Value: 90}` (Short: `\p{Nv=90}`) (5)
T `\p{Numeric_Value: 100}` (Short: `\p{Nv=100}`) (19)
T `\p{Numeric_Value: 200}` (Short: `\p{Nv=200}`) (2)
T `\p{Numeric_Value: 300}` (Short: `\p{Nv=300}`) (3)
T `\p{Numeric_Value: 400}` (Short: `\p{Nv=400}`) (2)
T `\p{Numeric_Value: 500}` (Short: `\p{Nv=500}`) (12)
T `\p{Numeric_Value: 600}` (Short: `\p{Nv=600}`) (2)
T `\p{Numeric_Value: 700}` (Short: `\p{Nv=700}`) (2)
T `\p{Numeric_Value: 800}` (Short: `\p{Nv=800}`) (2)
T `\p{Numeric_Value: 900}` (Short: `\p{Nv=900}`) (3)
T `\p{Numeric_Value: 1000}` (Short: `\p{Nv=1000}`) (16)
T `\p{Numeric_Value: 2000}` (Short: `\p{Nv=2000}`) (1)
T `\p{Numeric_Value: 3000}` (Short: `\p{Nv=3000}`) (1)
T `\p{Numeric_Value: 4000}` (Short: `\p{Nv=4000}`) (1)
T `\p{Numeric_Value: 5000}` (Short: `\p{Nv=5000}`) (5)
T `\p{Numeric_Value: 6000}` (Short: `\p{Nv=6000}`) (1)
T `\p{Numeric_Value: 7000}` (Short: `\p{Nv=7000}`) (1)
T `\p{Numeric_Value: 8000}` (Short: `\p{Nv=8000}`) (1)
T `\p{Numeric_Value: 9000}` (Short: `\p{Nv=9000}`) (1)
T `\p{Numeric_Value: 10000}` (= 1.0e+04) (Short: `\p{Nv=10000}`) (7)
T `\p{Numeric_Value: 20000}` (= 2.0e+04) (Short: `\p{Nv=20000}`) (1)
T `\p{Numeric_Value: 30000}` (= 3.0e+04) (Short: `\p{Nv=30000}`) (1)
T `\p{Numeric_Value: 40000}` (= 4.0e+04) (Short: `\p{Nv=40000}`) (1)
T `\p{Numeric_Value: 50000}` (= 5.0e+04) (Short: `\p{Nv=50000}`) (4)
T `\p{Numeric_Value: 60000}` (= 6.0e+04) (Short: `\p{Nv=60000}`) (1)
T `\p{Numeric_Value: 70000}` (= 7.0e+04) (Short: `\p{Nv=70000}`) (1)
T `\p{Numeric_Value: 80000}` (= 8.0e+04) (Short: `\p{Nv=80000}`) (1)
T `\p{Numeric_Value: 90000}` (= 9.0e+04) (Short: `\p{Nv=90000}`) (1)
T `\p{Numeric_Value: 100000}` (= 1.0e+05) (Short: `\p{Nv=100000}`) (1)

T	\p{Numeric_Value: 100000000} (= 1.0e+08) (Short: \p{Nv=100000000}) (2)
T	\p{Numeric_Value: 1000000000000} (= 1.0e+12) (Short: \p{Nv= 1000000000000}) (1)
	\p{Numeric_Value: NaN} (Short: \p{Nv=NaN}) (1_112_971)
D	\p{Nv: *}
D	\p{OAlpha}
	\p{Other_Alphabetic} (= \p{Other_Alphabetic=Y}) (759)
D	\p{OAlpha: *}
D	\p{ODI}
	\p{Other_Default_Ignorable_Code_Point} (= \p{Other_Default_Ignorable_Code_Point=Y}) (3778)
D	\p{ODI: *}
	\p{Other_Default_Ignorable_Code_Point: *}
	\p{Ogam}
	\p{Ogham} (= \p{Script=Ogham}) (NOT \p{Block=Ogham}) (29)
	\p{Ogham}
	\p{Script=Ogham} (Short: \p{Ogam}; NOT \p{Block=Ogham}) (29)
D	\p{OGr_Ext}
	\p{Other_Grapheme_Extend} (= \p{Other_Grapheme_Extend=Y}) (23)
D	\p{OGr_Ext: *}
D	\p{OIDC}
	\p{Other_ID_Continue} (= \p{Other_ID_Continue=Y}) (11)
D	\p{OIDC: *}
D	\p{OIDS}
	\p{Other_ID_Start} (= \p{Other_ID_Start=Y}) (4)
D	\p{OIDS: *}
	\p{Other_ID_Start: *}
	\p{Ol_Chiki}
	\p{Olck}
	\p{Old_Italic}
	\p{Script=Ol_Chiki} (Short: \p{Olck}) (48)
	\p{Ol_Chiki} (= \p{Script=Ol_Chiki}) (48)
	\p{Script=Old_Italic} (Short: \p{Ital}; NOT \p{Block=Old_Italic}) (35)
	\p{Old_Persian}
	\p{Script=Old_Persian} (Short: \p{Xpeo}; NOT \p{Block=Old_Persian}) (50)
	\p{Old_South_Arabian}
	\p{Script=Old_South_Arabian} (Short: \p{Sarab}) (32)
	\p{Old_Turkic}
	\p{Script=Old_Turkic} (Short: \p{Orkh}; NOT \p{Block=Old_Turkic}) (73)
D	\p{OLower}
	\p{Other_Lowercase} (= \p{Other_Lowercase=Y}) (159)
D	\p{OLower: *}
D	\p{OMath}
	\p{Other_Math} (= \p{Other_Math=Y}) (1216)
D	\p{OMath: *}
	\p{Open_Punctuation}
	\p{General_Category=Open_Punctuation} (Short: \p{Ps}) (72)
X	\p{Optical_Character_Recognition} \p{Block=Optical_Character_Recognition} (32)
	\p{Oriya}
	\p{Script=Oriya} (Short: \p{Orya}; NOT \p{Block=Oriya}) (84)
	\p{Orkh}
	\p{Old_Turkic} (= \p{Script=Old_Turkic}) (NOT \p{Block=Old_Turkic}) (73)
	\p{Orya}
	\p{Oriya} (= \p{Script=Oriya}) (NOT \p{Block=Oriya}) (84)
	\p{Osma}
	\p{Osmanya} (= \p{Script=Osmanya}) (NOT \p{Block=Osmanya}) (40)
	\p{Osmanya}
	\p{Script=Osmanya} (Short: \p{Osma}; NOT \p{Block=Osmanya}) (40)
	\p{Other}
	\p{General_Category=Other} (Short: \p{C})

- (1_006_956)
- D `\p{Other_Alphabetic}` `\p{Other_Alphabetic=Y}` (Short: `\p{OAlpha}`) (759)
- D `\p{Other_Alphabetic: N*}` Used by Unicode internally for generating the Alphabetic property (which should be used instead) and not intended to be used stand-alone (Short: `\p{OAlpha=N}`, `\P{OAlpha}`) (1_113_353)
- D `\p{Other_Alphabetic: Y*}` Used by Unicode internally for generating the Alphabetic property (which should be used instead) and not intended to be used stand-alone (Short: `\p{OAlpha=Y}`, `\p{OAlpha}`) (759)
- D `\p{Other_Default_Ignorable_Code_Point}`
`\p{Other_Default_Ignorable_Code_Point=Y}`
 (Short: `\p{ODI}`) (3778)
- D `\p{Other_Default_Ignorable_Code_Point: N*}` Used by Unicode internally for generating the Default_Ignorable_Code_Point property (which should be used instead) and not intended to be used stand-alone (Short: `\p{ODI=N}`, `\P{ODI}`) (1_110_334)
- D `\p{Other_Default_Ignorable_Code_Point: Y*}` Used by Unicode internally for generating the Default_Ignorable_Code_Point property (which should be used instead) and not intended to be used stand-alone (Short: `\p{ODI=Y}`, `\p{ODI}`) (3778)
- D `\p{Other_Grapheme_Extend}` `\p{Other_Grapheme_Extend=Y}` (Short: `\p{OGrExt}`) (23)
- D `\p{Other_Grapheme_Extend: N*}` Used by Unicode internally for generating the Grapheme_Extend property (which should be used instead) and not intended to be used stand-alone (Short: `\p{OGrExt=N}`, `\P{OGrExt}`) (1_114_089)
- D `\p{Other_Grapheme_Extend: Y*}` Used by Unicode internally for generating the Grapheme_Extend property (which should be used instead) and not intended to be used stand-alone (Short: `\p{OGrExt=Y}`, `\p{OGrExt}`) (23)
- D `\p{Other_ID_Continue}` `\p{Other_ID_Continue=Y}` (Short: `\p{OIDC}`) (11)
- D `\p{Other_ID_Continue: N*}` Used by Unicode internally for generating the ID_Continue property (which should be used instead) and not intended to be used stand-alone (Short: `\p{OIDC=N}`, `\P{OIDC}`) (1_114_101)
- D `\p{Other_ID_Continue: Y*}` Used by Unicode internally for generating the ID_Continue property (which should be used instead) and not intended to be used stand-alone (Short: `\p{OIDC=Y}`, `\p{OIDC}`) (11)
- D `\p{Other_ID_Start}` `\p{Other_ID_Start=Y}` (Short: `\p{OIDS}`) (4)
- D `\p{Other_ID_Start: N*}` Used by Unicode internally for generating the ID_Start property (which should be used instead) and not intended to be

	used stand-alone (Short: $\backslash\text{p}\{\text{OIDS}=\text{N}\}$, $\backslash\text{P}\{\text{OIDS}\}$) (1_114_108)
D $\backslash\text{p}\{\text{Other_ID_Start: Y}^*\}$	Used by Unicode internally for generating the ID_Start property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OIDS}=\text{Y}\}$, $\backslash\text{P}\{\text{OIDS}\}$) (4)
$\backslash\text{p}\{\text{Other_Letter}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Letter}\}$ (Short: $\backslash\text{p}\{\text{Lo}\}$) (96_128)
D $\backslash\text{p}\{\text{Other_Lowercase}\}$	$\backslash\text{p}\{\text{Other_Lowercase}=\text{Y}\}$ (Short: $\backslash\text{p}\{\text{OLower}\}$) (159)
D $\backslash\text{p}\{\text{Other_Lowercase: N}^*\}$	Used by Unicode internally for generating the Lowercase property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OLower}=\text{N}\}$, $\backslash\text{P}\{\text{OLower}\}$) (1_113_953)
D $\backslash\text{p}\{\text{Other_Lowercase: Y}^*\}$	Used by Unicode internally for generating the Lowercase property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OLower}=\text{Y}\}$, $\backslash\text{P}\{\text{OLower}\}$) (159)
D $\backslash\text{p}\{\text{Other_Math}\}$	$\backslash\text{p}\{\text{Other_Math}=\text{Y}\}$ (Short: $\backslash\text{p}\{\text{OMath}\}$) (1216)
D $\backslash\text{p}\{\text{Other_Math: N}^*\}$	Used by Unicode internally for generating the Math property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OMath}=\text{N}\}$, $\backslash\text{P}\{\text{OMath}\}$) (1_112_896)
D $\backslash\text{p}\{\text{Other_Math: Y}^*\}$	Used by Unicode internally for generating the Math property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OMath}=\text{Y}\}$, $\backslash\text{P}\{\text{OMath}\}$) (1216)
$\backslash\text{p}\{\text{Other_Number}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Number}\}$ (Short: $\backslash\text{p}\{\text{No}\}$) (429)
$\backslash\text{p}\{\text{Other_Punctuation}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Punctuation}\}$ (Short: $\backslash\text{p}\{\text{Po}\}$) (389)
$\backslash\text{p}\{\text{Other_Symbol}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Symbol}\}$ (Short: $\backslash\text{p}\{\text{So}\}$) (3409)
D $\backslash\text{p}\{\text{Other_Uppercase}\}$	$\backslash\text{p}\{\text{Other_Uppercase}=\text{Y}\}$ (Short: $\backslash\text{p}\{\text{OUpper}\}$) (42)
D $\backslash\text{p}\{\text{Other_Uppercase: N}^*\}$	Used by Unicode internally for generating the Uppercase property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OUpper}=\text{N}\}$, $\backslash\text{P}\{\text{OUpper}\}$) (1_114_070)
D $\backslash\text{p}\{\text{Other_Uppercase: Y}^*\}$	Used by Unicode internally for generating the Uppercase property (which should be used instead) and not intended to be used stand-alone (Short: $\backslash\text{p}\{\text{OUpper}=\text{Y}\}$, $\backslash\text{P}\{\text{OUpper}\}$) (42)
D $\backslash\text{p}\{\text{OUpper}\}$	$\backslash\text{p}\{\text{Other_Uppercase}\}$ (= $\backslash\text{p}\{\text{Other_Uppercase}=\text{Y}\}$) (42)
D $\backslash\text{p}\{\text{OUpper: }^*\}$	$\backslash\text{p}\{\text{Other_Uppercase: }^*\}$
$\backslash\text{p}\{\text{P}\}$	$\backslash\text{p}\{\text{Punct}\}$ (= $\backslash\text{p}\{\text{General_Category}=\text{Punctuation}\}$) (585)
$\backslash\text{p}\{\text{Paragraph_Separator}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Paragraph_Separator}\}$

	(Short: <code>\p{Zp}</code>) (1)
<code>\p{Pat_Syn}</code>	<code>\p{Pattern_Syntax}</code> (= <code>\p{Pattern_Syntax=Y}</code>) (2760)
<code>\p{Pat_Syn: *}</code>	<code>\p{Pattern_Syntax: *}</code>
<code>\p{Pat_WS}</code>	<code>\p{Pattern_White_Space}</code> (= <code>\p{Pattern_White_Space=Y}</code>) (11)
<code>\p{Pat_WS: *}</code>	<code>\p{Pattern_White_Space: *}</code>
<code>\p{Pattern_Syntax}</code>	<code>\p{Pattern_Syntax=Y}</code> (Short: <code>\p{PatSyn}</code>) (2760)
<code>\p{Pattern_Syntax: N*}</code>	(Short: <code>\p{PatSyn=N}</code> , <code>\P{PatSyn}</code>) (1_111_352)
<code>\p{Pattern_Syntax: Y*}</code>	(Short: <code>\p{PatSyn=Y}</code> , <code>\p{PatSyn}</code>) (2760)
<code>\p{Pattern_White_Space}</code>	<code>\p{Pattern_White_Space=Y}</code> (Short: <code>\p{PatWS}</code>) (11)
<code>\p{Pattern_White_Space: N*}</code>	(Short: <code>\p{PatWS=N}</code> , <code>\P{PatWS}</code>) (1_114_101)
<code>\p{Pattern_White_Space: Y*}</code>	(Short: <code>\p{PatWS=Y}</code> , <code>\p{PatWS}</code>) (11)
<code>\p{Pc}</code>	<code>\p{Connector_Punctuation}</code> (= <code>\p{General_Category=Connector_Punctuation}</code>) (10)
<code>\p{Pd}</code>	<code>\p{Dash_Punctuation}</code> (= <code>\p{General_Category=Dash_Punctuation}</code>) (21)
<code>\p{Pe}</code>	<code>\p{Close_Punctuation}</code> (= <code>\p{General_Category=Close_Punctuation}</code>) (71)
<code>\p{PerlSpace}</code>	<code>\s</code> , restricted to ASCII (5)
<code>\p{PerlWord}</code>	<code>\w</code> , restricted to ASCII = <code>[A-Za-z0-9_]</code> (63)
<code>\p{Pf}</code>	<code>\p{Final_Punctuation}</code> (= <code>\p{General_Category=Final_Punctuation}</code>) (10)
<code>\p{Phag}</code>	<code>\p{Phags_Pa}</code> (= <code>\p{Script=Phags_Pa}</code>) (NOT <code>\p{Block=Phags_Pa}</code>) (56)
<code>\p{Phags_Pa}</code>	<code>\p{Script=Phags_Pa}</code> (Short: <code>\p{Phag}</code> ; NOT <code>\p{Block=Phags_Pa}</code>) (56)
X <code>\p{Phaistos_Disc}</code>	<code>\p{Block=Phaistos_Disc}</code> (48)
<code>\p{Phli}</code>	<code>\p{Inscriptional_Pahlavi}</code> (= <code>\p{Script=Inscriptional_Pahlavi}</code>) (NOT <code>\p{Block=Inscriptional_Pahlavi}</code>) (27)
<code>\p{Phnx}</code>	<code>\p{Phoenician}</code> (= <code>\p{Script=Phoenician}</code>) (NOT <code>\p{Block=Phoenician}</code>) (29)
<code>\p{Phoenician}</code>	<code>\p{Script=Phoenician}</code> (Short: <code>\p{Phnx}</code> ; NOT <code>\p{Block=Phoenician}</code>) (29)
X <code>\p{Phonetic_Extensions}</code>	<code>\p{Block=Phonetic_Extensions}</code> (128)
X <code>\p{Phonetic_Extensions_Supplement}</code>	<code>\p{Block=Phonetic_Extensions_Supplement}</code> (64)
<code>\p{Pi}</code>	<code>\p{Initial_Punctuation}</code> (= <code>\p{General_Category=Initial_Punctuation}</code>) (12)
<code>\p{Po}</code>	<code>\p{Other_Punctuation}</code> (= <code>\p{General_Category=Other_Punctuation}</code>) (389)
<code>\p{PosixAlnum}</code>	<code>[A-Za-z0-9]</code> (62)
<code>\p{PosixAlpha}</code>	<code>[A-Za-z]</code> (52)
<code>\p{PosixBlank}</code>	<code>\t</code> and <code>' '</code> (2)
<code>\p{PosixCntrl}</code>	<code>[\x00-\x1F]</code> (33)

<code>\p{PosixDigit}</code>	<code>[0-9]</code> (10)
<code>\p{PosixGraph}</code>	<code>[\x21-\x7E]</code> (94)
<code>\p{PosixLower}</code>	<code>[a-z]</code> (26)
<code>\p{PosixPrint}</code>	<code>[\x20-\x7E]</code> (95)
<code>\p{PosixPunct}</code>	Graphical characters that aren't Word characters = <code>[\x21-\x2F\x3A-\x40\x5B-\x60\x7B-\x7E]</code> (32)
<code>\p{PosixSpace}</code>	<code>\t \n, \x0B, \f, \r, and ' '</code> (6)
<code>\p{PosixUpper}</code>	<code>[A-Z]</code> (26)
T <code>\p{Present_In: 1.1}</code>	<code>\p{Age=1.1}</code> (Short: <code>\p{In=1.1}</code>) (Perl extension) (33_979)
T <code>\p{Present_In: 2.0}</code>	Code point's usage introduced in version 2.0 or earlier (Short: <code>\p{In=2.0}</code>) (Perl extension) (178_500)
T <code>\p{Present_In: 2.1}</code>	Code point's usage introduced in version 2.1 or earlier (Short: <code>\p{In=2.1}</code>) (Perl extension) (178_502)
T <code>\p{Present_In: 3.0}</code>	Code point's usage introduced in version 3.0 or earlier (Short: <code>\p{In=3.0}</code>) (Perl extension) (188_809)
T <code>\p{Present_In: 3.1}</code>	Code point's usage introduced in version 3.1 or earlier (Short: <code>\p{In=3.1}</code>) (Perl extension) (233_787)
T <code>\p{Present_In: 3.2}</code>	Code point's usage introduced in version 3.2 or earlier (Short: <code>\p{In=3.2}</code>) (Perl extension) (234_803)
T <code>\p{Present_In: 4.0}</code>	Code point's usage introduced in version 4.0 or earlier (Short: <code>\p{In=4.0}</code>) (Perl extension) (236_029)
T <code>\p{Present_In: 4.1}</code>	Code point's usage introduced in version 4.1 or earlier (Short: <code>\p{In=4.1}</code>) (Perl extension) (237_302)
T <code>\p{Present_In: 5.0}</code>	Code point's usage introduced in version 5.0 or earlier (Short: <code>\p{In=5.0}</code>) (Perl extension) (238_671)
T <code>\p{Present_In: 5.1}</code>	Code point's usage introduced in version 5.1 or earlier (Short: <code>\p{In=5.1}</code>) (Perl extension) (240_295)
T <code>\p{Present_In: 5.2}</code>	Code point's usage introduced in version 5.2 or earlier (Short: <code>\p{In=5.2}</code>) (Perl extension) (246_943)
<code>\p{Present_In: Unassigned}</code>	<code>\p{Age=Unassigned}</code> (Short: <code>\p{In=Unassigned}</code>) (Perl extension) (867_169)
<code>\p{Print}</code>	Characters that are graphical plus space characters (but no controls) (244_762)
<code>\p{Private_Use}</code>	<code>\p{General_Category=Private_Use}</code> (Short: <code>\p{Co}</code> ; NOT <code>\p{Private_Use_Area}</code>) (137_468)
X <code>\p{Private_Use_Area}</code>	<code>\p{Block=Private_Use_Area}</code> (Short: <code>\p{InPrivateUse}</code>) (6400)
<code>\p{Prti}</code>	<code>\p{Inscriptional_Parthian}</code> (= <code>\p{Script=Inscriptional_Parthian}</code>) (NOT <code>\p{Block=Inscriptional_Parthian}</code>) (30)
<code>\p{Ps}</code>	<code>\p{Open_Punctuation}</code> (= <code>\p{General_Category=Open_Punctuation}</code>) (72)

<code>\p{Punct}</code>	<code>\p{General_Category=Punctuation}</code> (Short: <code>\p{P}</code>) (585)
<code>\p{Punctuation}</code>	<code>\p{Punct}</code> (= <code>\p{General_Category=Punctuation}</code>) (585)
<code>\p{Qaac}</code>	<code>\p{Coptic}</code> (= <code>\p{Script=Coptic}</code>) (NOT <code>\p{Block=Coptic}</code>) (135)
<code>\p{Qaai}</code>	<code>\p{Inherited}</code> (= <code>\p{Script=Inherited}</code>) (523)
<code>\p{QMark}</code>	<code>\p{Quotation_Mark}</code> (= <code>\p{Quotation_Mark=Y}</code>) (29)
<code>\p{QMark: *}</code>	<code>\p{Quotation_Mark: *}</code>
<code>\p{Quotation_Mark}</code>	<code>\p{Quotation_Mark=Y}</code> (Short: <code>\p{QMark}</code>) (29)
<code>\p{Quotation_Mark: N*}</code>	(Short: <code>\p{QMark=N}</code> , <code>\p{QMark}</code>) (1_114_083)
<code>\p{Quotation_Mark: Y*}</code>	(Short: <code>\p{QMark=Y}</code> , <code>\p{QMark}</code>) (29)
<code>\p{Radical}</code>	<code>\p{Radical=Y}</code> (329)
<code>\p{Radical: N*}</code>	(Single: <code>\p{Radical}</code>) (1_113_783)
<code>\p{Radical: Y*}</code>	(Single: <code>\p{Radical}</code>) (329)
<code>\p{Rejang}</code>	<code>\p{Script=Rejang}</code> (Short: <code>\p{Rjng}</code> ; NOT <code>\p{Block=Rejang}</code>) (37)
<code>\p{Rjng}</code>	<code>\p{Rejang}</code> (= <code>\p{Script=Rejang}</code>) (NOT <code>\p{Block=Rejang}</code>) (37)
X <code>\p{Rumi_Numeral_Symbols}</code>	<code>\p{Block=Rumi_Numeral_Symbols}</code> (32)
<code>\p{Runic}</code>	<code>\p{Script=Runic}</code> (Short: <code>\p{Runr}</code> ; NOT <code>\p{Block=Runic}</code>) (78)
<code>\p{Runr}</code>	<code>\p{Runic}</code> (= <code>\p{Script=Runic}</code>) (NOT <code>\p{Block=Runic}</code>) (78)
<code>\p{S}</code>	<code>\p{Symbol}</code> (= <code>\p{General_Category=Symbol}</code>) (4499)
<code>\p{Samaritan}</code>	<code>\p{Script=Samaritan}</code> (Short: <code>\p{Samr}</code> ; NOT <code>\p{Block=Samaritan}</code>) (61)
<code>\p{Samr}</code>	<code>\p{Samaritan}</code> (= <code>\p{Script=Samaritan}</code>) (NOT <code>\p{Block=Samaritan}</code>) (61)
<code>\p{Sarab}</code>	<code>\p{Old_South_Arabian}</code> (= <code>\p{Script=Old_South_Arabian}</code>) (32)
<code>\p{Saur}</code>	<code>\p{Saurashtra}</code> (= <code>\p{Script=Saurashtra}</code>) (NOT <code>\p{Block=Saurashtra}</code>) (81)
<code>\p{Saurashtra}</code>	<code>\p{Script=Saurashtra}</code> (Short: <code>\p{Saur}</code> ; NOT <code>\p{Block=Saurashtra}</code>) (81)
<code>\p{SB: *}</code>	<code>\p{Sentence_Break: *}</code>
<code>\p{Sc}</code>	<code>\p{Currency_Symbol}</code> (= <code>\p{General_Category=Currency_Symbol}</code>) (46)
<code>\p{Sc: *}</code>	<code>\p{Script: *}</code>
<code>\p{Script: Arab}</code>	<code>\p{Script=Arabic}</code> (1030)
<code>\p{Script: Arabic}</code>	(Short: <code>\p{Sc=Arab}</code> , <code>\p{Arab}</code>) (1030)
<code>\p{Script: Armenian}</code>	(Short: <code>\p{Sc=Armn}</code> , <code>\p{Armn}</code>) (90)
<code>\p{Script: Armi}</code>	<code>\p{Script=Imperial_Aramaic}</code> (31)
<code>\p{Script: Armn}</code>	<code>\p{Script=Armenian}</code> (90)
<code>\p{Script: Avestan}</code>	(Short: <code>\p{Sc=Avst}</code> , <code>\p{Avst}</code>) (61)
<code>\p{Script: Avst}</code>	<code>\p{Script=Avestan}</code> (61)
<code>\p{Script: Bali}</code>	<code>\p{Script=Balinese}</code> (121)
<code>\p{Script: Balinese}</code>	(Short: <code>\p{Sc=Bali}</code> , <code>\p{Bali}</code>) (121)
<code>\p{Script: Bamu}</code>	<code>\p{Script=Bamum}</code> (88)
<code>\p{Script: Bamum}</code>	(Short: <code>\p{Sc=Bamu}</code> , <code>\p{Bamu}</code>) (88)
<code>\p{Script: Beng}</code>	<code>\p{Script=Bengali}</code> (92)

<code>\p{Script: Bengali}</code>	(Short: <code>\p{Sc=Beng}</code> , <code>\p{Beng}</code>) (92)
<code>\p{Script: Bopo}</code>	<code>\p{Script=Bopomofo}</code> (65)
<code>\p{Script: Bopomofo}</code>	(Short: <code>\p{Sc=Bopo}</code> , <code>\p{Bopo}</code>) (65)
<code>\p{Script: Brai}</code>	<code>\p{Script=Braille}</code> (256)
<code>\p{Script: Braille}</code>	(Short: <code>\p{Sc=Brai}</code> , <code>\p{Brai}</code>) (256)
<code>\p{Script: Bugi}</code>	<code>\p{Script=Buginese}</code> (30)
<code>\p{Script: Buginese}</code>	(Short: <code>\p{Sc=Bugi}</code> , <code>\p{Bugi}</code>) (30)
<code>\p{Script: Buhd}</code>	<code>\p{Script=Buhid}</code> (20)
<code>\p{Script: Buhid}</code>	(Short: <code>\p{Sc=Buhd}</code> , <code>\p{Buhd}</code>) (20)
<code>\p{Script: Canadian_Aboriginal}</code>	(Short: <code>\p{Sc=Cans}</code> , <code>\p{Cans}</code>) (710)
<code>\p{Script: Cans}</code>	<code>\p{Script=Canadian_Aboriginal}</code> (710)
<code>\p{Script: Cari}</code>	<code>\p{Script=Carian}</code> (49)
<code>\p{Script: Carian}</code>	(Short: <code>\p{Sc=Cari}</code> , <code>\p{Cari}</code>) (49)
<code>\p{Script: Cham}</code>	(Short: <code>\p{Sc=Cham}</code> , <code>\p{Cham}</code>) (83)
<code>\p{Script: Cher}</code>	<code>\p{Script=Cherokee}</code> (85)
<code>\p{Script: Cherokee}</code>	(Short: <code>\p{Sc=Cher}</code> , <code>\p{Cher}</code>) (85)
<code>\p{Script: Common}</code>	(Short: <code>\p{Sc=Zyyy}</code> , <code>\p{Zyyy}</code>) (5395)
<code>\p{Script: Copt}</code>	<code>\p{Script=Coptic}</code> (135)
<code>\p{Script: Coptic}</code>	(Short: <code>\p{Sc=Copt}</code> , <code>\p{Copt}</code>) (135)
<code>\p{Script: Cpvt}</code>	<code>\p{Script=Cypriot}</code> (55)
<code>\p{Script: Cuneiform}</code>	(Short: <code>\p{Sc=Xsux}</code> , <code>\p{Xsux}</code>) (982)
<code>\p{Script: Cypriot}</code>	(Short: <code>\p{Sc=Cpvt}</code> , <code>\p{Cpvt}</code>) (55)
<code>\p{Script: Cyrillic}</code>	(Short: <code>\p{Sc=Cyrl}</code> , <code>\p{Cyrl}</code>) (404)
<code>\p{Script: Cyril}</code>	<code>\p{Script=Cyrillic}</code> (404)
<code>\p{Script: Deseret}</code>	(Short: <code>\p{Sc=Dsrt}</code> , <code>\p{Dsrt}</code>) (80)
<code>\p{Script: Deva}</code>	<code>\p{Script=Devanagari}</code> (140)
<code>\p{Script: Devanagari}</code>	(Short: <code>\p{Sc=Deva}</code> , <code>\p{Deva}</code>) (140)
<code>\p{Script: Dsrt}</code>	<code>\p{Script=Deseret}</code> (80)
<code>\p{Script: Egyp}</code>	<code>\p{Script=Egyptian_Hieroglyphs}</code> (1071)
<code>\p{Script: Egyptian_Hieroglyphs}</code>	(Short: <code>\p{Sc=Egyp}</code> , <code>\p{Egyp}</code>) (1071)
<code>\p{Script: Ethi}</code>	<code>\p{Script=Ethiopic}</code> (461)
<code>\p{Script: Ethiopic}</code>	(Short: <code>\p{Sc=Ethi}</code> , <code>\p{Ethi}</code>) (461)
<code>\p{Script: Geor}</code>	<code>\p{Script=Georgian}</code> (120)
<code>\p{Script: Georgian}</code>	(Short: <code>\p{Sc=Geor}</code> , <code>\p{Geor}</code>) (120)
<code>\p{Script: Glag}</code>	<code>\p{Script=Glagolitic}</code> (94)
<code>\p{Script: Glagolitic}</code>	(Short: <code>\p{Sc=Glag}</code> , <code>\p{Glag}</code>) (94)
<code>\p{Script: Goth}</code>	<code>\p{Script=Gothic}</code> (27)
<code>\p{Script: Gothic}</code>	(Short: <code>\p{Sc=Goth}</code> , <code>\p{Goth}</code>) (27)
<code>\p{Script: Greek}</code>	(Short: <code>\p{Sc=Grek}</code> , <code>\p{Grek}</code>) (511)
<code>\p{Script: Grek}</code>	<code>\p{Script=Greek}</code> (511)
<code>\p{Script: Gujarati}</code>	(Short: <code>\p{Sc=Gujr}</code> , <code>\p{Gujr}</code>) (83)
<code>\p{Script: Gujr}</code>	<code>\p{Script=Gujarati}</code> (83)
<code>\p{Script: Gurmukhi}</code>	(Short: <code>\p{Sc=Guru}</code> , <code>\p{Guru}</code>) (79)
<code>\p{Script: Guru}</code>	<code>\p{Script=Gurmukhi}</code> (79)
<code>\p{Script: Han}</code>	(Short: <code>\p{Sc=Han}</code> , <code>\p{Han}</code>) (75_738)
<code>\p{Script: Hang}</code>	<code>\p{Script=Hangul}</code> (11_737)
<code>\p{Script: Hangul}</code>	(Short: <code>\p{Sc=Hang}</code> , <code>\p{Hang}</code>) (11_737)
<code>\p{Script: Hani}</code>	<code>\p{Script=Han}</code> (75_738)
<code>\p{Script: Hano}</code>	<code>\p{Script=Hanunoo}</code> (21)
<code>\p{Script: Hanunoo}</code>	(Short: <code>\p{Sc=Hano}</code> , <code>\p{Hano}</code>) (21)
<code>\p{Script: Hebr}</code>	<code>\p{Script=Hebrew}</code> (133)
<code>\p{Script: Hebrew}</code>	(Short: <code>\p{Sc=Hebr}</code> , <code>\p{Hebr}</code>) (133)
<code>\p{Script: Hira}</code>	<code>\p{Script=Hiragana}</code> (90)
<code>\p{Script: Hiragana}</code>	(Short: <code>\p{Sc=Hira}</code> , <code>\p{Hira}</code>) (90)

```

\p{Script: Imperial_Aramaic} (Short: \p{Sc=Armi}, \p{Armi}) (31)
\p{Script: Inherited} (Short: \p{Sc=Zinh}, \p{Zinh}) (523)
\p{Script: Inscriptional_Pahlavi} (Short: \p{Sc=Phli}, \p{Phli})
(27)
\p{Script: Inscriptional_Parthian} (Short: \p{Sc=Prti}, \p{Prti})
(30)
\p{Script: Ital} \p{Script=Old_Italic} (35)
\p{Script: Java} \p{Script=Javanese} (91)
\p{Script: Javanese} (Short: \p{Sc=Java}, \p{Java}) (91)
\p{Script: Kaithi} (Short: \p{Sc=Kthi}, \p{Kthi}) (66)
\p{Script: Kali} \p{Script=Kayah_Li} (48)
\p{Script: Kana} \p{Script=Katakana} (299)
\p{Script: Kannada} (Short: \p{Sc=Knda}, \p{Knda}) (84)
\p{Script: Katakana} (Short: \p{Sc=Kana}, \p{Kana}) (299)
\p{Script: Kayah_Li} (Short: \p{Sc=Kali}, \p{Kali}) (48)
\p{Script: Khar} \p{Script=Kharoshthi} (65)
\p{Script: Kharoshthi} (Short: \p{Sc=Khar}, \p{Khar}) (65)
\p{Script: Khmer} (Short: \p{Sc=Khmr}, \p{Khmr}) (146)
\p{Script: Khmr} \p{Script=Khmer} (146)
\p{Script: Knda} \p{Script=Kannada} (84)
\p{Script: Kthi} \p{Script=Kaithi} (66)
\p{Script: Lana} \p{Script=Tai_Tham} (127)
\p{Script: Lao} (Short: \p{Sc=Lao}, \p{Lao}) (65)
\p{Script: Laoo} \p{Script=Lao} (65)
\p{Script: Latin} (Short: \p{Sc=Latn}, \p{Latn}) (1244)
\p{Script: Latn} \p{Script=Latin} (1244)
\p{Script: Lepc} \p{Script=Lepcha} (74)
\p{Script: Lepcha} (Short: \p{Sc=Lepc}, \p{Lepc}) (74)
\p{Script: Limb} \p{Script=Limbu} (66)
\p{Script: Limbu} (Short: \p{Sc=Limb}, \p{Limb}) (66)
\p{Script: Linb} \p{Script=Linear_B} (211)
\p{Script: Linear_B} (Short: \p{Sc=Linb}, \p{Linb}) (211)
\p{Script: Lisu} (Short: \p{Sc=Lisu}, \p{Lisu}) (48)
\p{Script: Lyci} \p{Script=Lycian} (29)
\p{Script: Lycian} (Short: \p{Sc=Lyci}, \p{Lyci}) (29)
\p{Script: Lydi} \p{Script=Lydian} (27)
\p{Script: Lydian} (Short: \p{Sc=Lydi}, \p{Lydi}) (27)
\p{Script: Malayalam} (Short: \p{Sc=Mlym}, \p{Mlym}) (95)
\p{Script: Meetei_Mayek} (Short: \p{Sc=Mtei}, \p{Mtei}) (56)
\p{Script: Mlym} \p{Script=Malayalam} (95)
\p{Script: Mong} \p{Script=Mongolian} (153)
\p{Script: Mongolian} (Short: \p{Sc=Mong}, \p{Mong}) (153)
\p{Script: Mtei} \p{Script=Meetei_Mayek} (56)
\p{Script: Myanmar} (Short: \p{Sc=Mymr}, \p{Mymr}) (188)
\p{Script: Mymr} \p{Script=Myanmar} (188)
\p{Script: New_Tai_Lue} (Short: \p{Sc= Talu}, \p{Talu}) (83)
\p{Script: Nko} (Short: \p{Sc=Nko}, \p{Nko}) (59)
\p{Script: Nkoo} \p{Script=Nko} (59)
\p{Script: Ogam} \p{Script=Ogham} (29)
\p{Script: Ogham} (Short: \p{Sc=Ogam}, \p{Ogam}) (29)
\p{Script: Ol_Chiki} (Short: \p{Sc=Olck}, \p{Olck}) (48)
\p{Script: Olck} \p{Script=Ol_Chiki} (48)
\p{Script: Old_Italic} (Short: \p{Sc=Ital}, \p{Ital}) (35)
\p{Script: Old_Persian} (Short: \p{Sc=Xpeo}, \p{Xpeo}) (50)
\p{Script: Old_South_Arabian} (Short: \p{Sc=Sarb}, \p{Sarb}) (32)
\p{Script: Old_Turkic} (Short: \p{Sc=Orkh}, \p{Orkh}) (73)

```

<code>\p{Script: Oriya}</code>	(Short: <code>\p{Sc=Orya}</code> , <code>\p{Orya}</code>) (84)
<code>\p{Script: Orkh}</code>	<code>\p{Script=Old_Turkic}</code> (73)
<code>\p{Script: Orya}</code>	<code>\p{Script=Oriya}</code> (84)
<code>\p{Script: Osma}</code>	<code>\p{Script=Osmanya}</code> (40)
<code>\p{Script: Osmanya}</code>	(Short: <code>\p{Sc=Osma}</code> , <code>\p{Osma}</code>) (40)
<code>\p{Script: Phag}</code>	<code>\p{Script=Phags_Pa}</code> (56)
<code>\p{Script: Phags_Pa}</code>	(Short: <code>\p{Sc=Phag}</code> , <code>\p{Phag}</code>) (56)
<code>\p{Script: Phli}</code>	<code>\p{Script=Inscriptional_Pahlavi}</code> (27)
<code>\p{Script: Phnx}</code>	<code>\p{Script=Phoenician}</code> (29)
<code>\p{Script: Phoenician}</code>	(Short: <code>\p{Sc=Phnx}</code> , <code>\p{Phnx}</code>) (29)
<code>\p{Script: Prti}</code>	<code>\p{Script=Inscriptional_Parthian}</code> (30)
<code>\p{Script: Qaac}</code>	<code>\p{Script=Coptic}</code> (135)
<code>\p{Script: Qaai}</code>	<code>\p{Script=Inherited}</code> (523)
<code>\p{Script: Rejang}</code>	(Short: <code>\p{Sc=Rjng}</code> , <code>\p{Rjng}</code>) (37)
<code>\p{Script: Rjng}</code>	<code>\p{Script=Rejang}</code> (37)
<code>\p{Script: Runic}</code>	(Short: <code>\p{Sc=Runr}</code> , <code>\p{Runr}</code>) (78)
<code>\p{Script: Runr}</code>	<code>\p{Script=Runic}</code> (78)
<code>\p{Script: Samaritan}</code>	(Short: <code>\p{Sc=Samr}</code> , <code>\p{Samr}</code>) (61)
<code>\p{Script: Samr}</code>	<code>\p{Script=Samaritan}</code> (61)
<code>\p{Script: Sarb}</code>	<code>\p{Script=Old_South_Arabian}</code> (32)
<code>\p{Script: Saur}</code>	<code>\p{Script=Saurashtra}</code> (81)
<code>\p{Script: Saurashtra}</code>	(Short: <code>\p{Sc=Saur}</code> , <code>\p{Saur}</code>) (81)
<code>\p{Script: Shavian}</code>	(Short: <code>\p{Sc=Shaw}</code> , <code>\p{Shaw}</code>) (48)
<code>\p{Script: Shav}</code>	<code>\p{Script=Shavian}</code> (48)
<code>\p{Script: Sinh}</code>	<code>\p{Script=Sinhala}</code> (80)
<code>\p{Script: Sinhala}</code>	(Short: <code>\p{Sc=Sinh}</code> , <code>\p{Sinh}</code>) (80)
<code>\p{Script: Sund}</code>	<code>\p{Script=Sundanese}</code> (55)
<code>\p{Script: Sundanese}</code>	(Short: <code>\p{Sc=Sund}</code> , <code>\p{Sund}</code>) (55)
<code>\p{Script: Sylo}</code>	<code>\p{Script=Syloti_Nagri}</code> (44)
<code>\p{Script: Syloti_Nagri}</code>	(Short: <code>\p{Sc=Sylo}</code> , <code>\p{Sylo}</code>) (44)
<code>\p{Script: Syrc}</code>	<code>\p{Script=Syriac}</code> (77)
<code>\p{Script: Syriac}</code>	(Short: <code>\p{Sc=Syrc}</code> , <code>\p{Syrc}</code>) (77)
<code>\p{Script: Tagalog}</code>	(Short: <code>\p{Sc=Tglg}</code> , <code>\p{Tglg}</code>) (20)
<code>\p{Script: Tagb}</code>	<code>\p{Script=Tagbanwa}</code> (18)
<code>\p{Script: Tagbanwa}</code>	(Short: <code>\p{Sc=Tagb}</code> , <code>\p{Tagb}</code>) (18)
<code>\p{Script: Tai_Le}</code>	(Short: <code>\p{Sc=Tale}</code> , <code>\p{Tale}</code>) (35)
<code>\p{Script: Tai_Tham}</code>	(Short: <code>\p{Sc=Lana}</code> , <code>\p{Lana}</code>) (127)
<code>\p{Script: Tai_Viet}</code>	(Short: <code>\p{Sc=Tavt}</code> , <code>\p{Tavt}</code>) (72)
<code>\p{Script: Tale}</code>	<code>\p{Script=Tai_Le}</code> (35)
<code>\p{Script: Talu}</code>	<code>\p{Script=New_Tai_Lue}</code> (83)
<code>\p{Script: Tamil}</code>	(Short: <code>\p{Sc=Taml}</code> , <code>\p{Taml}</code>) (72)
<code>\p{Script: Taml}</code>	<code>\p{Script=Tamil}</code> (72)
<code>\p{Script: Tavt}</code>	<code>\p{Script=Tai_Viet}</code> (72)
<code>\p{Script: Telu}</code>	<code>\p{Script=Telugu}</code> (93)
<code>\p{Script: Telugu}</code>	(Short: <code>\p{Sc=Telu}</code> , <code>\p{Telu}</code>) (93)
<code>\p{Script: Tfng}</code>	<code>\p{Script=Tifinagh}</code> (55)
<code>\p{Script: Tglg}</code>	<code>\p{Script=Tagalog}</code> (20)
<code>\p{Script: Thaa}</code>	<code>\p{Script=Thaana}</code> (50)
<code>\p{Script: Thaana}</code>	(Short: <code>\p{Sc=Thaa}</code> , <code>\p{Thaa}</code>) (50)
<code>\p{Script: Thai}</code>	(Short: <code>\p{Sc=Thai}</code> , <code>\p{Thai}</code>) (86)
<code>\p{Script: Tibetan}</code>	(Short: <code>\p{Sc=Tibt}</code> , <code>\p{Tibt}</code>) (201)
<code>\p{Script: Tibt}</code>	<code>\p{Script=Tibetan}</code> (201)
<code>\p{Script: Tifinagh}</code>	(Short: <code>\p{Sc=Tfng}</code> , <code>\p{Tfng}</code>) (55)
<code>\p{Script: Ugar}</code>	<code>\p{Script=Ugaritic}</code> (31)
<code>\p{Script: Ugaritic}</code>	(Short: <code>\p{Sc=Ugar}</code> , <code>\p{Ugar}</code>) (31)
<code>\p{Script: Unknown}</code>	(Short: <code>\p{Sc=Zzzz}</code> , <code>\p{Zzzz}</code>) (1_006_751)

<code>\p{Script: Vai}</code>	<code>(Short: \p{Sc=Vai}, \p{Vai})</code> (300)
<code>\p{Script: Vaii}</code>	<code>\p{Script=Vai}</code> (300)
<code>\p{Script: Xpeo}</code>	<code>\p{Script=Old_Persian}</code> (50)
<code>\p{Script: Xsux}</code>	<code>\p{Script=Cuneiform}</code> (982)
<code>\p{Script: Yi}</code>	<code>(Short: \p{Sc=Yi}, \p{Yi})</code> (1220)
<code>\p{Script: Yiii}</code>	<code>\p{Script=Yi}</code> (1220)
<code>\p{Script: Zinh}</code>	<code>\p{Script=Inherited}</code> (523)
<code>\p{Script: Zyyy}</code>	<code>\p{Script=Common}</code> (5395)
<code>\p{Script: Zzzz}</code>	<code>\p{Script=Unknown}</code> (1_006_751)
<code>\p{SD}</code>	<code>\p{Soft_Dotted}</code> (= <code>\p{Soft_Dotted=Y}</code>) (46)
<code>\p{SD: *}</code>	<code>\p{Soft_Dotted: *}</code>
<code>\p{Sentence_Break: AT}</code>	<code>\p{Sentence_Break=ATerm}</code> (4)
<code>\p{Sentence_Break: ATerm}</code>	<code>(Short: \p{SB=AT})</code> (4)
<code>\p{Sentence_Break: CL}</code>	<code>\p{Sentence_Break=Close}</code> (177)
<code>\p{Sentence_Break: Close}</code>	<code>(Short: \p{SB=CL})</code> (177)
<code>\p{Sentence_Break: CR}</code>	<code>(Short: \p{SB=CR})</code> (1)
<code>\p{Sentence_Break: EX}</code>	<code>\p{Sentence_Break=Extend}</code> (1455)
<code>\p{Sentence_Break: Extend}</code>	<code>(Short: \p{SB=EX})</code> (1455)
<code>\p{Sentence_Break: FO}</code>	<code>\p{Sentence_Break=Format}</code> (138)
<code>\p{Sentence_Break: Format}</code>	<code>(Short: \p{SB=FO})</code> (138)
<code>\p{Sentence_Break: LE}</code>	<code>\p{Sentence_Break=OLetter}</code> (96_405)
<code>\p{Sentence_Break: LF}</code>	<code>(Short: \p{SB=LF})</code> (1)
<code>\p{Sentence_Break: LO}</code>	<code>\p{Sentence_Break=Lower}</code> (1907)
<code>\p{Sentence_Break: Lower}</code>	<code>(Short: \p{SB=LO})</code> (1907)
<code>\p{Sentence_Break: NU}</code>	<code>\p{Sentence_Break=Numeric}</code> (403)
<code>\p{Sentence_Break: Numeric}</code>	<code>(Short: \p{SB=NU})</code> (403)
<code>\p{Sentence_Break: OLetter}</code>	<code>(Short: \p{SB=LE})</code> (96_405)
<code>\p{Sentence_Break: Other}</code>	<code>(Short: \p{SB=XX})</code> (1_012_008)
<code>\p{Sentence_Break: SC}</code>	<code>\p{Sentence_Break=SContinue}</code> (26)
<code>\p{Sentence_Break: SContinue}</code>	<code>(Short: \p{SB=SC})</code> (26)
<code>\p{Sentence_Break: SE}</code>	<code>\p{Sentence_Break=Sep}</code> (3)
<code>\p{Sentence_Break: Sep}</code>	<code>(Short: \p{SB=SE})</code> (3)
<code>\p{Sentence_Break: Sp}</code>	<code>(Short: \p{SB=Sp})</code> (21)
<code>\p{Sentence_Break: ST}</code>	<code>\p{Sentence_Break=STerm}</code> (63)
<code>\p{Sentence_Break: STerm}</code>	<code>(Short: \p{SB=ST})</code> (63)
<code>\p{Sentence_Break: UP}</code>	<code>\p{Sentence_Break=Upper}</code> (1500)
<code>\p{Sentence_Break: Upper}</code>	<code>(Short: \p{SB=UP})</code> (1500)
<code>\p{Sentence_Break: XX}</code>	<code>\p{Sentence_Break=Other}</code> (1_012_008)
<code>\p{Separator}</code>	<code>\p{General_Category=Separator}</code> (Short: <code>\p{Z}</code>) (20)
<code>\p{Shavian}</code>	<code>\p{Script=Shavian}</code> (Short: <code>\p{Shaw}</code>) (48)
<code>\p{Shaw}</code>	<code>\p{Shavian}</code> (= <code>\p{Script=Shavian}</code>) (48)
<code>\p{Sinh}</code>	<code>\p{Sinhala}</code> (= <code>\p{Script=Sinhala}</code>) (NOT <code>\p{Block=Sinhala}</code>) (80)
<code>\p{Sinhala}</code>	<code>\p{Script=Sinhala}</code> (Short: <code>\p{Sinh}</code> ; NOT <code>\p{Block=Sinhala}</code>) (80)
<code>\p{Sk}</code>	<code>\p{Modifier_Symbol}</code> (= <code>\p{General_Category=Modifier_Symbol}</code>) (99)
<code>\p{Sm}</code>	<code>\p{Math_Symbol}</code> (= <code>\p{General_Category=Math_Symbol}</code>) (945)
X <code>\p{Small_Form_Variants}</code>	<code>\p{Block=Small_Form_Variants}</code> (32)
<code>\p{So}</code>	<code>\p{Other_Symbol}</code> (= <code>\p{General_Category=Other_Symbol}</code>) (3409)
<code>\p{Soft_Dotted}</code>	<code>\p{Soft_Dotted=Y}</code> (Short: <code>\p{SD}</code>) (46)
<code>\p{Soft_Dotted: N*}</code>	<code>(Short: \p{SD=N}, \p{SD})</code> (1_114_066)

<code>\p{Soft_Dotted: Y*}</code>	<code>(Short: \p{SD=Y}, \p{SD})</code> (46)
<code>\p{Space}</code>	<code>\p{White_Space=Y}</code> \s including beyond ASCII plus vertical tab (26)
<code>\p{Space: *}</code>	<code>\p{White_Space: *}</code>
<code>\p{Space_Separator}</code>	<code>\p{General_Category=Space_Separator}</code> (Short: <code>\p{Zs}</code>) (18)
<code>\p{SpacePerl}</code>	\s, including beyond ASCII (25)
<code>\p{Spacing_Mark}</code>	<code>\p{General_Category=Spacing_Mark}</code> (Short: <code>\p{Mc}</code>) (276)
X <code>\p{Spacing_Modifier_Letters}</code>	<code>\p{Block=Spacing_Modifier_Letters}</code> (80)
X <code>\p{Specials}</code>	<code>\p{Block=Specials}</code> (16)
<code>\p{STerm}</code>	<code>\p{STerm=Y}</code> (66)
<code>\p{STerm: N*}</code>	(Single: <code>\p{STerm}</code>) (1_114_046)
<code>\p{STerm: Y*}</code>	(Single: <code>\p{STerm}</code>) (66)
<code>\p{Sund}</code>	<code>\p{Sundanese}</code> (= <code>\p{Script=Sundanese}</code>) (NOT <code>\p{Block=Sundanese}</code>) (55)
<code>\p{Sundanese}</code>	<code>\p{Script=Sundanese}</code> (Short: <code>\p{Sund}</code> ; NOT <code>\p{Block=Sundanese}</code>) (55)
X <code>\p{Superscripts_And_Subscripts}</code>	<code>\p{Block=Superscripts_And_Subscripts}</code> (48)
X <code>\p{Supplemental_Arrows_A}</code>	<code>\p{Block=Supplemental_Arrows_A}</code> (16)
X <code>\p{Supplemental_Arrows_B}</code>	<code>\p{Block=Supplemental_Arrows_B}</code> (128)
X <code>\p{Supplemental_Mathematical_Operators}</code>	<code>\p{Block=Supplemental_Mathematical_Operators}</code> (256)
X <code>\p{Supplemental_Punctuation}</code>	<code>\p{Block=Supplemental_Punctuation}</code> (128)
X <code>\p{Supplementary_Private_Use_Area_A}</code>	<code>\p{Block=Supplementary_Private_Use_Area_A}</code> (65_536)
X <code>\p{Supplementary_Private_Use_Area_B}</code>	<code>\p{Block=Supplementary_Private_Use_Area_B}</code> (65_536)
<code>\p{Surrogate}</code>	<code>\p{General_Category=Surrogate}</code> (Short: <code>\p{Cs}</code>) (2048)
<code>\p{Sylo}</code>	<code>\p{Syloti_Nagri}</code> (= <code>\p{Script=Syloti_Nagri}</code>) (NOT <code>\p{Block=Syloti_Nagri}</code>) (44)
<code>\p{Syloti_Nagri}</code>	<code>\p{Script=Syloti_Nagri}</code> (Short: <code>\p{Sylo}</code> ; NOT <code>\p{Block=Syloti_Nagri}</code>) (44)
<code>\p{Symbol}</code>	<code>\p{General_Category=Symbol}</code> (Short: <code>\p{S}</code>) (4499)
<code>\p{Syrc}</code>	<code>\p{Syriac}</code> (= <code>\p{Script=Syriac}</code>) (NOT <code>\p{Block=Syriac}</code>) (77)
<code>\p{Syriac}</code>	<code>\p{Script=Syriac}</code> (Short: <code>\p{Syrc}</code> ; NOT <code>\p{Block=Syriac}</code>) (77)
<code>\p{Tagalog}</code>	<code>\p{Script=Tagalog}</code> (Short: <code>\p{Tglg}</code> ; NOT <code>\p{Block=Tagalog}</code>) (20)
<code>\p{Tagb}</code>	<code>\p{Tagbanwa}</code> (= <code>\p{Script=Tagbanwa}</code>) (NOT <code>\p{Block=Tagbanwa}</code>) (18)
<code>\p{Tagbanwa}</code>	<code>\p{Script=Tagbanwa}</code> (Short: <code>\p{Tagb}</code> ; NOT <code>\p{Block=Tagbanwa}</code>) (18)
X <code>\p{Tags}</code>	<code>\p{Block=Tags}</code> (128)
<code>\p{Tai_Le}</code>	<code>\p{Script=Tai_Le}</code> (Short: <code>\p{Tale}</code> ; NOT <code>\p{Block=Tai_Le}</code>) (35)

<code>\p{Tai_Tham}</code>	<code>\p{Script=Tai_Tham}</code> (Short: <code>\p{Lana}</code> ; NOT <code>\p{Block=Tai_Tham}</code>) (127)
<code>\p{Tai_Viet}</code>	<code>\p{Script=Tai_Viet}</code> (Short: <code>\p{Tavt}</code> ; NOT <code>\p{Block=Tai_Viet}</code>) (72)
X <code>\p{Tai_Xuan_Jing_Symbols}</code>	<code>\p{Block=Tai_Xuan_Jing_Symbols}</code> (96)
<code>\p{Tale}</code>	<code>\p{Tai_Le}</code> (= <code>\p{Script=Tai_Le}</code>) (NOT <code>\p{Block=Tai_Le}</code>) (35)
<code>\p{Talu}</code>	<code>\p{New_Tai_Lue}</code> (= <code>\p{Script=New_Tai_Lue}</code>) (NOT <code>\p{Block=New_Tai_Lue}</code>) (83)
<code>\p{Tamil}</code>	<code>\p{Script=Tamil}</code> (Short: <code>\p{Taml}</code> ; NOT <code>\p{Block=Tamil}</code>) (72)
<code>\p{Taml}</code>	<code>\p{Tamil}</code> (= <code>\p{Script=Tamil}</code>) (NOT <code>\p{Block=Tamil}</code>) (72)
<code>\p{Tavt}</code>	<code>\p{Tai_Viet}</code> (= <code>\p{Script=Tai_Viet}</code>) (NOT <code>\p{Block=Tai_Viet}</code>) (72)
<code>\p{Telu}</code>	<code>\p{Telugu}</code> (= <code>\p{Script=Telugu}</code>) (NOT <code>\p{Block=Telugu}</code>) (93)
<code>\p{Telugu}</code>	<code>\p{Script=Telugu}</code> (Short: <code>\p{Telu}</code> ; NOT <code>\p{Block=Telugu}</code>) (93)
<code>\p{Term}</code>	<code>\p{Terminal_Punctuation}</code> (= <code>\p{Terminal_Punctuation=Y}</code>) (161)
<code>\p{Term: *}</code>	<code>\p{Terminal_Punctuation: *}</code>
<code>\p{Terminal_Punctuation}</code>	<code>\p{Terminal_Punctuation=Y}</code> (Short: <code>\p{Term}</code>) (161)
<code>\p{Terminal_Punctuation: N*}</code>	(Short: <code>\p{Term=N}</code> , <code>\p{Term}</code>) (1_113_951)
<code>\p{Terminal_Punctuation: Y*}</code>	(Short: <code>\p{Term=Y}</code> , <code>\p{Term}</code>) (161)
<code>\p{Tfng}</code>	<code>\p{Tifinagh}</code> (= <code>\p{Script=Tifinagh}</code>) (NOT <code>\p{Block=Tifinagh}</code>) (55)
<code>\p{Tglg}</code>	<code>\p{Tagalog}</code> (= <code>\p{Script=Tagalog}</code>) (NOT <code>\p{Block=Tagalog}</code>) (20)
<code>\p{Thaa}</code>	<code>\p{Thaana}</code> (= <code>\p{Script=Thaana}</code>) (NOT <code>\p{Block=Thaana}</code>) (50)
<code>\p{Thaana}</code>	<code>\p{Script=Thaana}</code> (Short: <code>\p{Thaa}</code> ; NOT <code>\p{Block=Thaana}</code>) (50)
<code>\p{Thai}</code>	<code>\p{Script=Thai}</code> (NOT <code>\p{Block=Thai}</code>) (86)
<code>\p{Tibetan}</code>	<code>\p{Script=Tibetan}</code> (Short: <code>\p{Tibt}</code> ; NOT <code>\p{Block=Tibetan}</code>) (201)
<code>\p{Tibt}</code>	<code>\p{Tibetan}</code> (= <code>\p{Script=Tibetan}</code>) (NOT <code>\p{Block=Tibetan}</code>) (201)
<code>\p{Tifinagh}</code>	<code>\p{Script=Tifinagh}</code> (Short: <code>\p{Tfng}</code> ; NOT <code>\p{Block=Tifinagh}</code>) (55)
<code>\p>Title</code>	<code>\p{General_Category=Titlecase_Letter}</code> (Short: <code>\p{Lt}</code>) (31)
<code>\p>Titlecase_Letter}</code>	<code>\p>Title</code> (= <code>\p{General_Category=Titlecase_Letter}</code>) (31)
<code>\p{Ugar}</code>	<code>\p{Ugaritic}</code> (= <code>\p{Script=Ugaritic}</code>) (NOT <code>\p{Block=Ugaritic}</code>) (31)
<code>\p{Ugaritic}</code>	<code>\p{Script=Ugaritic}</code> (Short: <code>\p{Ugar}</code> ; NOT <code>\p{Block=Ugaritic}</code>) (31)
<code>\p{UIdeo}</code>	<code>\p{Unified_Ideograph}</code> (= <code>\p{Unified_Ideograph=Y}</code>) (74_394)
<code>\p{UIdeo: *}</code>	<code>\p{Unified_Ideograph: *}</code>
<code>\p{Unassigned}</code>	<code>\p{General_Category=Unassigned}</code> (Short: <code>\p{Cn}</code>) (867_235)
X <code>\p{Unified_Canadian_Aboriginal_Syllabics}</code>	<code>\p{Block=</code>

	Unified_Canadian_Aboriginal_Syllabics}
	(Short: <code>\p{InCanadianSyllabics}</code>) (640)
X <code>\p{Unified_Canadian_Aboriginal_Syllabics_Extended}</code>	<code>\p{Block=Unified_Canadian_Aboriginal_Syllabics_Extended}</code> (80)
<code>\p{Unified_Ideograph}</code>	<code>\p{Unified_Ideograph=Y}</code> (Short: <code>\p{UIdeo}</code>) (74_394)
<code>\p{Unified_Ideograph: N*}</code>	(Short: <code>\p{UIdeo=N}</code> , <code>\P{UIdeo}</code>) (1_039_718)
<code>\p{Unified_Ideograph: Y*}</code>	(Short: <code>\p{UIdeo=Y}</code> , <code>\p{UIdeo}</code>) (74_394)
<code>\p{Unknown}</code>	<code>\p{Script=Unknown}</code> (Short: <code>\p{Zzzz}</code>) (1_006_751)
<code>\p{Upper}</code>	<code>\p{Uppercase=Y}</code> (1469)
<code>\p{Upper: *}</code>	<code>\p{Uppercase: *}</code>
<code>\p{Uppercase}</code>	<code>\p{Upper}</code> (= <code>\p{Uppercase=Y}</code>) (1469)
<code>\p{Uppercase: N*}</code>	(Short: <code>\p{Upper=N}</code> , <code>\P{Upper}</code>) (1_112_643)
<code>\p{Uppercase: Y*}</code>	(Short: <code>\p{Upper=Y}</code> , <code>\p{Upper}</code>) (1469)
<code>\p{Uppercase_Letter}</code>	<code>\p{General_Category=Uppercase_Letter}</code> (Short: <code>\p{Lu}</code>) (1427)
<code>\p{Vai}</code>	<code>\p{Script=Vai}</code> (NOT <code>\p{Block=Vai}</code>) (300)
<code>\p{Vaii}</code>	<code>\p{Vai}</code> (= <code>\p{Script=Vai}</code>) (NOT <code>\p{Block=Vai}</code>) (300)
<code>\p{Variation_Selector}</code>	<code>\p{Variation_Selector=Y}</code> (Short: <code>\p{VS}</code>) (259)
<code>\p{Variation_Selector: N*}</code>	(Short: <code>\p{VS=N}</code> , <code>\P{VS}</code>) (1_113_853)
<code>\p{Variation_Selector: Y*}</code>	(Short: <code>\p{VS=Y}</code> , <code>\p{VS}</code>) (259)
X <code>\p{Variation_Selectors}</code>	<code>\p{Block=Variation_Selectors}</code> (16)
X <code>\p{Variation_Selectors_Supplement}</code>	<code>\p{Block=Variation_Selectors_Supplement}</code> (240)
X <code>\p{Vedic_Extensions}</code>	<code>\p{Block=Vedic_Extensions}</code> (48)
X <code>\p{Vertical_Forms}</code>	<code>\p{Block=Vertical_Forms}</code> (16)
<code>\p{VertSpace}</code>	<code>\v</code> (7)
<code>\p{VS}</code>	<code>\p{Variation_Selector}</code> (= <code>\p{Variation_Selector=Y}</code>) (259)
<code>\p{VS: *}</code>	<code>\p{Variation_Selector: *}</code>
<code>\p{WB: *}</code>	<code>\p{Word_Break: *}</code>
<code>\p{White_Space}</code>	<code>\p{White_Space=Y}</code> (Short: <code>\p{WSpace}</code>) (26)
<code>\p{White_Space: N*}</code>	(Short: <code>\p{Space=N}</code> , <code>\P{WSpace}</code>) (1_114_086)
<code>\p{White_Space: Y*}</code>	(Short: <code>\p{Space=Y}</code> , <code>\p{WSpace}</code>) (26)
<code>\p{Word}</code>	<code>\w</code> , including beyond ASCII (101_685)
<code>\p{Word_Break: ALetter}</code>	(Short: <code>\p{WB=LE}</code>) (23_694)
<code>\p{Word_Break: CR}</code>	(Short: <code>\p{WB=CR}</code>) (1)
<code>\p{Word_Break: EX}</code>	<code>\p{Word_Break=ExtendNumLet}</code> (10)
<code>\p{Word_Break: Extend}</code>	(Short: <code>\p{WB=Extend}</code>) (1455)
<code>\p{Word_Break: ExtendNumLet}</code>	(Short: <code>\p{WB=EX}</code>) (10)
<code>\p{Word_Break: FO}</code>	<code>\p{Word_Break=Format}</code> (137)
<code>\p{Word_Break: Format}</code>	(Short: <code>\p{WB=FO}</code>) (137)
<code>\p{Word_Break: KA}</code>	<code>\p{Word_Break=Katakana}</code> (309)
<code>\p{Word_Break: Katakana}</code>	(Short: <code>\p{WB=KA}</code>) (309)
<code>\p{Word_Break: LE}</code>	<code>\p{Word_Break=ALetter}</code> (23_694)
<code>\p{Word_Break: LF}</code>	(Short: <code>\p{WB=LF}</code>) (1)
<code>\p{Word_Break: MB}</code>	<code>\p{Word_Break=MidNumLet}</code> (8)
<code>\p{Word_Break: MidLetter}</code>	(Short: <code>\p{WB=ML}</code>) (8)
<code>\p{Word_Break: MidNum}</code>	(Short: <code>\p{WB=MN}</code>) (15)
<code>\p{Word_Break: MidNumLet}</code>	(Short: <code>\p{WB=MB}</code>) (8)

<code>\p{Word_Break: ML}</code>	<code>\p{Word_Break=MidLetter}</code> (8)
<code>\p{Word_Break: MN}</code>	<code>\p{Word_Break=MidNum}</code> (15)
<code>\p{Word_Break: Newline}</code>	(Short: <code>\p{WB=NL}</code>) (5)
<code>\p{Word_Break: NL}</code>	<code>\p{Word_Break=Newline}</code> (5)
<code>\p{Word_Break: NU}</code>	<code>\p{Word_Break=Numeric}</code> (402)
<code>\p{Word_Break: Numeric}</code>	(Short: <code>\p{WB=NU}</code>) (402)
<code>\p{Word_Break: Other}</code>	(Short: <code>\p{WB=XX}</code>) (1_088_067)
<code>\p{Word_Break: XX}</code>	<code>\p{Word_Break=Other}</code> (1_088_067)
<code>\p{WSpace}</code>	<code>\p{White_Space}</code> (= <code>\p{White_Space=Y}</code>) (26)
<code>\p{WSpace: *}</code>	<code>\p{White_Space: *}</code>
<code>\p{XDigit}</code>	<code>\p{Hex_Digit=Y}</code> (Short: <code>\p{Hex}</code>) (44)
<code>\p{XID_Continue}</code>	<code>\p{XID_Continue=Y}</code> (Short: <code>\p{XIDC}</code>) (101_615)
<code>\p{XID_Continue: N*}</code>	(Short: <code>\p{XIDC=N}</code> , <code>\p{XIDC}</code>) (1_012_497)
<code>\p{XID_Continue: Y*}</code>	(Short: <code>\p{XIDC=Y}</code> , <code>\p{XIDC}</code>) (101_615)
<code>\p{XID_Start}</code>	<code>\p{XID_Start=Y}</code> (Short: <code>\p{XIDS}</code>) (99_741)
<code>\p{XID_Start: N*}</code>	(Short: <code>\p{XIDS=N}</code> , <code>\p{XIDS}</code>) (1_014_371)
<code>\p{XID_Start: Y*}</code>	(Short: <code>\p{XIDS=Y}</code> , <code>\p{XIDS}</code>) (99_741)
<code>\p{XIDC}</code>	<code>\p{XID_Continue}</code> (= <code>\p{XID_Continue=Y}</code>) (101_615)
<code>\p{XIDC: *}</code>	<code>\p{XID_Continue: *}</code>
<code>\p{XIDS}</code>	<code>\p{XID_Start}</code> (= <code>\p{XID_Start=Y}</code>) (99_741)
<code>\p{XIDS: *}</code>	<code>\p{XID_Start: *}</code>
<code>\p{Xpeo}</code>	<code>\p{Old_Persian}</code> (= <code>\p{Script=Old_Persian}</code>) (NOT <code>\p{Block=Old_Persian}</code>) (50)
<code>\p{Xsux}</code>	<code>\p{Cuneiform}</code> (= <code>\p{Script=Cuneiform}</code>) (NOT <code>\p{Block=Cuneiform}</code>) (982)
<code>\p{Yi}</code>	<code>\p{Script=Yi}</code> (1220)
X <code>\p{Yi_Radicals}</code>	<code>\p{Block=Yi_Radicals}</code> (64)
X <code>\p{Yi_Syllables}</code>	<code>\p{Block=Yi_Syllables}</code> (1168)
<code>\p{Yiii}</code>	<code>\p{Yi}</code> (= <code>\p{Script=Yi}</code>) (1220)
X <code>\p{Yijing_Hexagram_Symbols}</code>	<code>\p{Block=Yijing_Hexagram_Symbols}</code> (64)
<code>\p{Z}</code>	<code>\p{Separator}</code> (= <code>\p{General_Category=Separator}</code>) (20)
<code>\p{Zinh}</code>	<code>\p{Inherited}</code> (= <code>\p{Script=Inherited}</code>) (523)
<code>\p{Zl}</code>	<code>\p{Line_Separator}</code> (= <code>\p{General_Category=Line_Separator}</code>) (1)
<code>\p{Zp}</code>	<code>\p{Paragraph_Separator}</code> (= <code>\p{General_Category=Paragraph_Separator}</code>) (1)
<code>\p{Zs}</code>	<code>\p{Space_Separator}</code> (= <code>\p{General_Category=Space_Separator}</code>) (18)
<code>\p{Zyyy}</code>	<code>\p{Common}</code> (= <code>\p{Script=Common}</code>) (5395)
<code>\p{Zzzz}</code>	<code>\p{Unknown}</code> (= <code>\p{Script=Unknown}</code>) (1_006_751)
T <code>\p{_\CanonDCIJ}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Soft_Dotted=Y}</code>) (46)
T <code>\p{_\Case_Ignorable}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Case_Ignorable=Y}</code>) (1632)
T <code>\p{_\CombAbove}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Canonical_Combining_Class=Above}</code>) (318)
T <code>\p{_\X_Begin}</code>	(For internal use by Perl, not necessarily stable) (1_113_907)

T \p{ <i>_X_Extend</i> }	(For internal use by Perl, not necessarily stable) (1462)
T \p{ <i>_X_LV_LVT_V</i> }	(For internal use by Perl, not necessarily stable) (11_267)

Legal \p{} and \P{} constructs that match no characters

Unicode has some property-value pairs that currently don't match anything. This happens generally either because they are obsolete, or for symmetry with other forms, but no language has yet been encoded that uses them. In this version of Unicode, the following match zero code points:

```
\p{Canonical_Combining_Class=Attached_Below_Left}
\p{Joining_Type=Left_Joining}
```

Properties not accessible through \p{} and \P{}

A few properties are accessible in Perl via various function calls only. These are: Lowercase_Mapping lc() and lcfirst() Titlecase_Mapping ucfirst() Uppercase_Mapping uc()

Case_Folding is accessible through the /i modifier in regular expressions.

The Name property is accessible through the \N{} interpolation in double-quoted strings and regular expressions, but both usages require a use *charnames*; to be specified, which also contains related functions *viacode()* and *vianame()*.

Unicode regular expression properties that are NOT accepted by Perl

Perl will generate an error for a few character properties in Unicode when used in a regular expression. The non-Unihan ones are listed below, with the reasons they are not accepted, perhaps with work-arounds. The short names for the properties are listed enclosed in (parentheses).

Expands_On_NFC (XO_NFC)

Expands_On_NFD (XO_NFD)

Expands_On_NFKC (XO_NFKC)

Expands_On_NFKD (XO_NFKD)

Easily computed, and yet doesn't cover the common encoding forms (UTF-16/8)

Grapheme_Link (Gr_Link)

Deprecated by Unicode. Use *ccc=vr* (Canonical_Combining_Class=Virama) instead

Jamo_Short_Name (JSN)

Used by Unicode internally for generating other properties and not intended to be used stand-alone

Script=Katakana_Or_Hiragana (sc=Hrkt)

Obsolete. All code points previously matched by this have been moved to "Script=Common"

An installation can choose to allow any of these to be matched by changing the controlling lists

contained in the program `$Config{privlib}/unicore/lib/unicore/mktables` and then re-running `lib/unicore/mktables`. (`%Config` is available from the `Config` module).

Files in the `To` directory (for serious hackers only)

All Unicode properties are really mappings (in the mathematical sense) from code points to their respective values. As part of its build process, Perl constructs tables containing these mappings for all properties that it deals with. But only a few of these are written out into files. Those written out are in the directory `$Config{privlib}/unicore/To/` (`%Config` is available from the `Config` module).

Those ones written are ones needed by Perl internally during execution, or for which there is some demand, and those for which there is no access through the Perl core. Generally, properties that can be used in regular expression matching do not have their map tables written, like `Script`. Nor are the simplistic properties that have a better, more complete version, such as `Simple_Uppercase_Mapping` (`Uppercase_Mapping` is written instead).

None of the properties in the `To` directory are currently directly accessible through the Perl core, although some may be accessed indirectly. For example, the `uc()` function implements the `Uppercase_Mapping` property and uses the `Upper.pl` file found in this directory.

The available files with their properties (short names in parentheses), and any flags or comments about them, are:

<code>Bmg.pl</code>	<code>Bidi_Mirroring_Glyph (bmg)</code>
<code>Digit.pl</code>	<code>Perl_Decimal_Digit</code>
<code>Fold.pl</code>	<code>Case_Folding (cf)</code>
<code>Lower.pl</code>	<code>Lowercase_Mapping (lc)</code>
<code>NFKCCF.pl</code>	<code>NFKC_Casefold (NFKC_CF)</code>
<code>Title.pl</code>	<code>Titlecase_Mapping (tc)</code>
<code>Upper.pl</code>	<code>Uppercase_Mapping (uc)</code>

An installation can choose to change which files are generated by changing the controlling lists contained in the program `$Config{privlib}/unicore/lib/unicore/mktables` and then re-running `lib/unicore/mktables`.

Each of these files defines two hash entries to help reading programs decipher it. One of them looks like this:

```
$utf8::SwashInfo{'ToNAME'}{'format'} = 's';
```

where `'NAME'` is a name to indicate the property. For backwards compatibility, this is not necessarily the property's official Unicode name. (The `'To'` is also for backwards compatibility.) The hash entry gives the format of the mapping fields of the table, currently one of the following:

<code>b</code>	binary
<code>d</code>	single decimal digit
<code>f</code>	floating point number
<code>i</code>	integer
<code>r</code>	rational: an integer or a fraction
<code>s</code>	arbitrary string
<code>x</code>	positive hex whole number; a code point

This format applies only to the entries in the main body of the table. Entries defined in hashes or ones that are missing from the list can have a different format.

The value that the missing entries have is given by the other `SwashInfo` hash entry line; it looks like this:

```
$utf8::SwashInfo{'ToNAME'}{'missing'} = 'NaN';
```

This example line says that any Unicode code points not explicitly listed in the file have the value 'NaN' under the property indicated by NAME. If the value is the special string `<code point>`, it means that the value for any missing code point is the code point itself. This happens, for example, in the file for Uppercase_Mapping (To/Upper.pl), in which code points like the character 'A', are missing because the uppercase of 'A' is itself.

SEE ALSO

<http://www.unicode.org/reports/tr44/>

perlrecharclass

perlunicode