

!!!!!!! DO NOT EDIT THIS FILE !!!!!!! # This file is machine-generated by lib/unicore/mktables from the Unicode # database, Version 6.0.0. Any changes made here will be lost!

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

NAME

perluniprops - Index of Unicode Version 6.0.0 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 "Other Properties" in *perlunicode*.

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}` and `\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 by Unicode, but still available. There are several varieties of obsolescence:

Obsolete

Properties marked with an 'O' in the table are considered obsolete.

Stabilized

Obsolete properties may be stabilized. Such a determination does not indicate that the property should or should not be used; instead it is a declaration that the property will not be maintained nor extended for newly encoded characters. Such properties are marked with an 'S' in the table.

Deprecated

An obsolete property may be deprecated, perhaps because its original intent has been replaced by another property, or because its specification was somehow defective. This means that its 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 preceded 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.

Most properties match the same code points regardless of whether `/i` case-insensitive matching is specified or not. But a few properties are affected. These are shown with the notation

```
(/i= other_property)
```

in the second column. Under case-insensitive matching they match the same code points as the property "other_property".

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 \p{Age: 3.1}	Code point's usage was introduced in version 3.1; See also Property 'Present_In' (44_978)
T \p{Age: 3.2}	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)
T \p{Age: 6.0}	Code point's usage was introduced in version 6.0; See also Property 'Present_In' (2088)
\p{Age: Unassigned}	Code point's usage has not been assigned in any Unicode release thus far. (865_081)
\p{AHex}	\p{ASCII_Hex_Digit} (= \p{ASCII_Hex_Digit= Y}) (22)
\p{AHex: *}	\p{ASCII_Hex_Digit: *}
X \p{Alchemical_Symbols}	\p{Block=Alchemical_Symbols} (128)
\p{All}	\p{Any} (1_114_112)
\p{Alnum}	Alphabetic and (Decimal) Numeric (101_959)
\p{Alpha}	\p{Alphabetic=Y} (101_539)
\p{Alpha: *}	\p{Alphabetic: *}
\p{Alphabetic}	\p{Alpha} (= \p{Alphabetic=Y}) (101_539)
\p{Alphabetic: N*}	(Short: \p{Alpha=N}, \p{Alpha}) (1_012_573)
\p{Alphabetic: Y*}	(Short: \p{Alpha=Y}, \p{Alpha}) (101_539)
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}) (1051)
\p{Arabic}	\p{Script=Arabic} (Short: \p{Arab}; NOT \p{Block=Arabic}) (1051)
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)

<code>\p{Armenian}</code>	<code>\p{Script=Armenian}</code> (Short: <code>\p{Armn}</code> ; NOT <code>\p{Block=Armenian}</code>) (90)
<code>\p{Armi}</code>	<code>\p{Imperial_Aramaic}</code> (= <code>\p{Script=Imperial_Aramaic}</code>) (NOT <code>\p{Block=Imperial_Aramaic}</code>) (31)
<code>\p{Armn}</code>	<code>\p{Armenian}</code> (= <code>\p{Script=Armenian}</code>) (NOT <code>\p{Block=Armenian}</code>) (90)
X <code>\p{Arrows}</code>	<code>\p{Block=Arrows}</code> (112)
<code>\p{ASCII}</code>	<code>\p{Block=Basic_Latin}</code> <code>[[[:ASCII:]]</code> (128)
<code>\p{ASCII_Hex_Digit}</code>	<code>\p{ASCII_Hex_Digit=Y}</code> (Short: <code>\p{AHex}</code>) (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 (248_965)
<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>) (657)
<code>\p{Bamum}</code>	<code>\p{Script=Bamum}</code> (Short: <code>\p{Bamu}</code> ; NOT <code>\p{Block=Bamum}</code>) (657)
X <code>\p{Bamum_Supplement}</code>	<code>\p{Block=Bamum_Supplement}</code> (576)
X <code>\p{Basic_Latin}</code>	<code>\p{ASCII}</code> (= <code>\p{Block=Basic_Latin}</code>) (128)
<code>\p{Batak}</code>	<code>\p{Script=Batak}</code> (Short: <code>\p{Batk}</code> ; NOT <code>\p{Block=Batak}</code>) (56)
<code>\p{Batk}</code>	<code>\p{Batak}</code> (= <code>\p{Script=Batak}</code>) (NOT <code>\p{Block=Batak}</code>) (56)
<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> (4015)
<code>\p{Bidi_Class: Boundary_Neutral}</code>	(Short: <code>\p{Bc=BN}</code>) (4015)
<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> (64)
<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>) (64)
<code>\p{Bidi_Class: L}</code>	<code>\p{Bidi_Class=Left_To_Right}</code> (1_098_619)
<code>\p{Bidi_Class: Left_To_Right}</code>	(Short: <code>\p{Bc=L}</code>) (1_098_619)

```

\p{Bidi_Class: Left_To_Right_Embedding} (Short: \p{Bc=LRE}) (1)
\p{Bidi_Class: Left_To_Right_Override} (Short: \p{Bc=LRO}) (1)
\p{Bidi_Class: LRE} \p{Bidi_Class=Left_To_Right_Embedding} (1)
\p{Bidi_Class: LRO} \p{Bidi_Class=Left_To_Right_Override} (1)
\p{Bidi_Class: Nonspacing_Mark} (Short: \p{Bc=NSM}) (1209)
\p{Bidi_Class: NSM} \p{Bidi_Class=Nonspacing_Mark} (1209)
\p{Bidi_Class: ON} \p{Bidi_Class=Other_Neutral} (4412)
\p{Bidi_Class: Other_Neutral} (Short: \p{Bc=ON}) (4412)
\p{Bidi_Class: Paragraph_Separator} (Short: \p{Bc=B}) (7)
\p{Bidi_Class: PDF} \p{Bidi_Class=Pop_Directional_Format} (1)
\p{Bidi_Class: Pop_Directional_Format} (Short: \p{Bc=PDF}) (1)
\p{Bidi_Class: R} \p{Bidi_Class=Right_To_Left} (4438)
\p{Bidi_Class: Right_To_Left} (Short: \p{Bc=R}) (4438)
\p{Bidi_Class: Right_To_Left_Embedding} (Short: \p{Bc=RLE}) (1)
\p{Bidi_Class: Right_To_Left_Override} (Short: \p{Bc=RLO}) (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: Alchemical_Symbols} (Single: \p{InAlchemicalSymbols})
(128)
\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: Bamum_Supplement}` (Single: `\p{InBamumSupplement}`) (576)
`\p{Block: Basic_Latin}` (Short: `\p{Blk=ASCII}`, `\p{ASCII}`) (128)
`\p{Block: Batak}` (Single: `\p{InBatak}`; NOT `\p{Batak}` NOR
`\p{Is_Batak}`) (64)
`\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: Brahmi}` (Single: `\p{InBrahmi}`; NOT `\p{Brahmi}` NOR
`\p{Is_Brahmi}`) (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: CJK_Unified_Ideographs_Extension_D} (Single:
    \p{InCJKUnifiedIdeographsExtensionD})
(224)
\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: Emoticons} (Single: \p{InEmoticons}) (80)
\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_Extended_A} (Single: \p{InEthiopicExtendedA})
        (48)
\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: Kana_Supplement} (Single: \p{InKanaSupplement}) (256)
\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: Mandaic} (Single: \p{InMandaic}; NOT \p{Mandaic}
NOR \p{Is_Mandaic}) (32)
\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_Symbols_And_Pictographs} (Single:
\p{InMiscellaneousSymbolsAnd-
Pictographs}) (768)
\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}) (861_664)
\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: Playing_Cards} (Single: \p{InPlayingCards}) (96)
\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)
\p{Block: Runic} (Single: \p{InRunic}; NOT \p{Runic} NOR
\p{Is_Runic}) (96)
\p{Block: Samaritan} (Single: \p{InSamaritan}; NOT
\p{Samaritan} NOR \p{Is_Samaritan}) (64)
\p{Block: Saurashtra} (Single: \p{InSaurashtra}; NOT
\p{Saurashtra} NOR \p{Is_Saurashtra})
(96)
\p{Block: Shavian} (Single: \p{InShavian}) (48)
\p{Block: Sinhala} (Single: \p{InSinhala}; NOT \p{Sinhala}
NOR \p{Is_Sinhala}) (128)
\p{Block: Small_Form_Variants} (Single: \p{InSmallFormVariants})
(32)
\p{Block: Spacing_Modifier_Letters} (Single:
\p{InSpacingModifierLetters}) (80)
\p{Block: Specials} (Single: \p{InSpecials}) (16)
\p{Block: Sundanese} (Single: \p{InSundanese}; NOT
\p{Sundanese} NOR \p{Is_Sundanese}) (64)
\p{Block: Superscripts_And_Subscripts} (Single:
\p{InSuperscriptsAndSubscripts}) (48)
\p{Block: Supplemental_Arrows_A} (Single:
\p{InSupplementalArrowsA}) (16)

```

```

\p{Block: Supplemental_Arrows_B} (Single:
    \p{InSupplementalArrowsB}) (128)
\p{Block: Supplemental_Mathematical_Operators} (Single:
    \p{InSupplementalMathematicalOperators})
    (256)
\p{Block: Supplemental_Punctuation} (Single:
    \p{InSupplementalPunctuation}) (128)
\p{Block: Supplementary_Private_Use_Area_A} (Single:
    \p{InSupplementaryPrivateUseAreaA})
    (65_536)
\p{Block: Supplementary_Private_Use_Area_B} (Single:
    \p{InSupplementaryPrivateUseAreaB})
    (65_536)
\p{Block: Syloti_Nagri} (Single: \p{InSylotiNagri}; NOT
    \p{Syloti_Nagri} NOR
    \p{Is_Syloti_Nagri}) (48)
\p{Block: Syriac} (Single: \p{InSyriac}; NOT \p{Syriac} NOR
    \p{Is_Syriac}) (80)
\p{Block: Tagalog} (Single: \p{InTagalog}; NOT \p{Tagalog}
    NOR \p{Is_Tagalog}) (32)
\p{Block: Tagbanwa} (Single: \p{InTagbanwa}; NOT \p{Tagbanwa}
    NOR \p{Is_Tagbanwa}) (32)
\p{Block: Tags} (Single: \p{InTags}) (128)
\p{Block: Tai_Le} (Single: \p{InTaiLe}; NOT \p{Tai_Le} NOR
    \p{Is_Tai_Le}) (48)
\p{Block: Tai_Tham} (Single: \p{InTaiTham}; NOT \p{Tai_Tham}
    NOR \p{Is_Tai_Tham}) (144)
\p{Block: Tai_Viet} (Single: \p{InTaiViet}; NOT \p{Tai_Viet}
    NOR \p{Is_Tai_Viet}) (96)
\p{Block: Tai_Xuan_Jing_Symbols} (Single:
    \p{InTaiXuanJingSymbols}) (96)
\p{Block: Tamil} (Single: \p{InTamil}; NOT \p{Tamil} NOR
    \p{Is_Tamil}) (128)
\p{Block: Telugu} (Single: \p{InTelugu}; NOT \p{Telugu} NOR
    \p{Is_Telugu}) (128)
\p{Block: Thaana} (Single: \p{InThaana}; NOT \p{Thaana} NOR
    \p{Is_Thaana}) (64)
\p{Block: Thai} (Single: \p{InThai}; NOT \p{Thai} NOR
    \p{Is_Thai}) (128)
\p{Block: Tibetan} (Single: \p{InTibetan}; NOT \p{Tibetan}
    NOR \p{Is_Tibetan}) (256)
\p{Block: Tifinagh} (Single: \p{InTifinagh}; NOT \p{Tifinagh}
    NOR \p{Is_Tifinagh}) (80)
\p{Block: Transport_And_Map_Symbols} (Single:
    \p{InTransportAndMapSymbols}) (128)
\p{Block: Ugaritic} (Single: \p{InUgaritic}; NOT \p{Ugaritic}
    NOR \p{Is_Ugaritic}) (32)
\p{Block: Unified_Canadian_Aboriginal_Syllabics} (Short: \p{Blk=
    CanadianSyllabics},
    \p{InCanadianSyllabics}) (640)
\p{Block: Unified_Canadian_Aboriginal_Syllabics_Extended} (Single:
    \p{InUnifiedCanadianAboriginalSyllabics-
    Extended}) (80)
\p{Block: Vai} (Single: \p{InVai}; NOT \p{Vai} NOR
    \p{Is_Vai}) (320)
\p{Block: Variation_Selectors} (Single: \p{InVariationSelectors})

```

	(16)
	<code>\p{Block: Variation_Selectors_Supplement}</code> (Single: <code>\p{InVariationSelectorsSupplement}</code>) (240)
	<code>\p{Block: Vedic_Extensions}</code> (Single: <code>\p{InVedicExtensions}</code>) (48)
	<code>\p{Block: Vertical_Forms}</code> (Single: <code>\p{InVerticalForms}</code>) (16)
	<code>\p{Block: Yi_Radicals}</code> (Single: <code>\p{InYiRadicals}</code>) (64)
	<code>\p{Block: Yi_Syllables}</code> (Single: <code>\p{InYiSyllables}</code>) (1168)
	<code>\p{Block: Yijing_Hexagram_Symbols}</code> (Single: <code>\p{InYijingHexagramSymbols}</code>) (64)
X	<code>\p{Block_Elements}</code> <code>\p{Block=Block_Elements}</code> (32)
	<code>\p{Bopo}</code> <code>\p{Bopomofo}</code> (= <code>\p{Script=Bopomofo}</code>) (NOT <code>\p{Block=Bopomofo}</code>) (70)
	<code>\p{Bopomofo}</code> <code>\p{Script=Bopomofo}</code> (Short: <code>\p{Bopo}</code> ; NOT <code>\p{Block=Bopomofo}</code>) (70)
X	<code>\p{Bopomofo_Extended}</code> <code>\p{Block=Bopomofo_Extended}</code> (32)
X	<code>\p{Box_Drawing}</code> <code>\p{Block=Box_Drawing}</code> (128)
	<code>\p{Brah}</code> <code>\p{Brahmi}</code> (= <code>\p{Script=Brahmi}</code>) (NOT <code>\p{Block=Brahmi}</code>) (108)
	<code>\p{Brahmi}</code> <code>\p{Script=Brahmi}</code> (Short: <code>\p{Brah}</code> ; NOT <code>\p{Block=Brahmi}</code>) (108)
	<code>\p{Brai}</code> <code>\p{Braille}</code> (= <code>\p{Script=Braille}</code>) (256)
	<code>\p{Braille}</code> <code>\p{Script=Braille}</code> (Short: <code>\p{Brai}</code>) (256)
X	<code>\p{Braille_Patterns}</code> <code>\p{Block=Braille_Patterns}</code> (256)
	<code>\p{Bugi}</code> <code>\p{Buginese}</code> (= <code>\p{Script=Buginese}</code>) (NOT <code>\p{Block=Buginese}</code>) (30)
	<code>\p{Buginese}</code> <code>\p{Script=Buginese}</code> (Short: <code>\p{Bugi}</code> ; NOT <code>\p{Block=Buginese}</code>) (30)
	<code>\p{Buhd}</code> <code>\p{Buhid}</code> (= <code>\p{Script=Buhid}</code>) (NOT <code>\p{Block=Buhid}</code>) (20)
	<code>\p{Buhid}</code> <code>\p{Script=Buhid}</code> (Short: <code>\p{Buhd}</code> ; NOT <code>\p{Block=Buhid}</code>) (20)
X	<code>\p{Byzantine_Musical_Symbols}</code> <code>\p{Block=Byzantine_Musical_Symbols}</code> (256)
	<code>\p{C}</code> <code>\p{Other}</code> (= <code>\p{General_Category=Other}</code>) (1_004_868)
	<code>\p{Canadian_Aboriginal}</code> <code>\p{Script=Canadian_Aboriginal}</code> (Short: <code>\p{Cans}</code>) (710)
X	<code>\p{Canadian_Syllabics}</code> <code>\p{Unified_Canadian_Aboriginal_Syllabics}</code> (= <code>\p{Block=Unified_Canadian_Aboriginal_Syllabics}</code>) (640)
T	<code>\p{Canonical_Combining_Class: 0}</code> <code>\p{Canonical_Combining_Class=Not_Reordered}</code> (1_113_506)
T	<code>\p{Canonical_Combining_Class: 1}</code> <code>\p{Canonical_Combining_Class=Overlay}</code> (26)
T	<code>\p{Canonical_Combining_Class: 7}</code> <code>\p{Canonical_Combining_Class=Nukta}</code> (12)
T	<code>\p{Canonical_Combining_Class: 8}</code> <code>\p{Canonical_Combining_Class=Kana_Voicing}</code> (2)
T	<code>\p{Canonical_Combining_Class: 9}</code> <code>\p{Canonical_Combining_Class=Virama}</code> (31)
T	<code>\p{Canonical_Combining_Class: 10}</code> (Short: <code>\p{Ccc=10}</code>) (1)
T	<code>\p{Canonical_Combining_Class: 11}</code> (Short: <code>\p{Ccc=11}</code>) (1)
T	<code>\p{Canonical_Combining_Class: 12}</code> (Short: <code>\p{Ccc=12}</code>) (1)
T	<code>\p{Canonical_Combining_Class: 13}</code> (Short: <code>\p{Ccc=13}</code>) (1)
T	<code>\p{Canonical_Combining_Class: 14}</code> (Short: <code>\p{Ccc=14}</code>) (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} (121)
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} (320)
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} (4)
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} (320)
\p{Canonical_Combining_Class: Above} (Short: \p{Ccc=A}) (320)
\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} (121)
\p{Canonical_Combining_Class: Below} (Short: \p{Ccc=B}) (121)
\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} (4)
\p{Canonical_Combining_Class: Double_Above} (Short: \p{Ccc=DA}) (5)
\p{Canonical_Combining_Class: Double_Below} (Short: \p{Ccc=DB}) (4)
\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} (12)
\p{Canonical_Combining_Class: Not_Reordered} (Short: \p{Ccc=NR})
        (1_113_506)
\p{Canonical_Combining_Class: NR} \p{Canonical_Combining_Class=
        Not_Reordered} (1_113_506)
\p{Canonical_Combining_Class: Nukta} (Short: \p{Ccc=NK}) (12)
\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}) (31)
\p{Canonical_Combining_Class: VR} \p{Canonical_Combining_Class=
        Virama} (31)
\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}) (1692)
\p{Case_Ignorable: N*} (Short: \p{CI=N}, \p{CI}) (1_112_420)
\p{Case_Ignorable: Y*} (Short: \p{CI=Y}, \p{CI}) (1692)
\p{Cased}         \p{Cased=Y} (3427)
\p{Cased: N*}     (Single: \p{Cased}) (1_110_685)
\p{Cased: Y*}     (Single: \p{Cased}) (3427)
\p{Cased_Letter} \p{General_Category=Cased_Letter} (Short:
        \p{LC}) (3226)
\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{CWCF}) (1102)
\p{Changes_When_Casefolded: N*} (Short: \p{CWCF=N}, \p{CWCF})
        (1_113_010)
\p{Changes_When_Casefolded: Y*} (Short: \p{CWCF=Y}, \p{CWCF})
        (1102)
\p{Changes_When_Casemapped} \p{Changes_When_Casemapped=Y} (Short:
        \p{CWCM}) (2128)
\p{Changes_When_Casemapped: N*} (Short: \p{CWCM=N}, \p{CWCM})
        (1_111_984)
\p{Changes_When_Casemapped: Y*} (Short: \p{CWCM=Y}, \p{CWCM})
        (2128)
\p{Changes_When_Lowercased} \p{Changes_When_Lowercased=Y} (Short:
        \p{CWL}) (1038)
\p{Changes_When_Lowercased: N*} (Short: \p{CWL=N}, \p{CWL})
        (1_113_074)
\p{Changes_When_Lowercased: Y*} (Short: \p{CWL=Y}, \p{CWL}) (1038)

```

```

\p{Changes_When_NFKC_Casefolded} \p{Changes_When_NFKC_Casefolded=
    Y} (Short: \p{CWKCF}) (9792)
\p{Changes_When_NFKC_Casefolded: N*} (Short: \p{CWKCF=N},
    \p{CWKCF}) (1_104_320)
\p{Changes_When_NFKC_Casefolded: Y*} (Short: \p{CWKCF=Y},
    \p{CWKCF}) (9792)
\p{Changes_When_Titlecased} \p{Changes_When_Titlecased=Y} (Short:
    \p{CWT}) (1094)
\p{Changes_When_Titlecased: N*} (Short: \p{CWT=N}, \p{CWT})
    (1_113_018)
\p{Changes_When_Titlecased: Y*} (Short: \p{CWT=Y}, \p{CWT}) (1094)
\p{Changes_When_Uppercased} \p{Changes_When_Uppercased=Y} (Short:
    \p{CWU}) (1121)
\p{Changes_When_Uppercased: N*} (Short: \p{CWU=N}, \p{CWU})
    (1_112_991)
\p{Changes_When_Uppercased: Y*} (Short: \p{CWU=Y}, \p{CWU}) (1121)
\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}) (1692)
\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)
X \p{CJK_Unified_Ideographs_Extension_D} \p{Block=
    CJK_Unified_Ideographs_Extension_D} (224)
\p{Close_Punctuation} \p{General_Category=Close_Punctuation}
    (Short: \p{Pe}) (71)
\p{Cn} \p{Unassigned} (= \p{General_Category=
    Unassigned}) (865_147)
\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	<code>\p{Combining_Diacritical_Marks}</code>	<code>\p{Block=Combining_Diacritical_Marks}</code>	(112)
X	<code>\p{Combining_Diacritical_Marks_For_Symbols}</code>	<code>\p{Block=Combining_Diacritical_Marks_For_Symbols}</code> (Short: <code>\p{InCombiningMarksForSymbols}</code>)	(48)
X	<code>\p{Combining_Diacritical_Marks_Supplement}</code>	<code>\p{Block=Combining_Diacritical_Marks_Supplement}</code>	(64)
X	<code>\p{Combining_Half_Marks}</code>	<code>\p{Block=Combining_Half_Marks}</code>	(16)
X	<code>\p{Combining_Marks_For_Symbols}</code>	<code>\p{Combining_Diacritical_Marks_For_Symbols}</code> (= <code>\p{Block=Combining_Diacritical_Marks_For_Symbols}</code>)	(48)
	<code>\p{Common}</code>	<code>\p{Script=Common}</code>	(Short: <code>\p{Zyyy}</code>) (6379)
X	<code>\p{Common_Indic_Number_Forms}</code>	<code>\p{Block=Common_Indic_Number_Forms}</code>	(16)
	<code>\p{Comp_Ex}</code>	<code>\p{Full_Composition_Exclusion}</code> (= <code>\p{Full_Composition_Exclusion=Y}</code>)	(1118)
	<code>\p{Comp_Ex: *}</code>	<code>\p{Full_Composition_Exclusion: *}</code>	
	<code>\p{Composition_Exclusion}</code>	<code>\p{Composition_Exclusion=Y}</code>	(Short: <code>\p{CE}</code>) (81)
	<code>\p{Composition_Exclusion: N*}</code>	(Short: <code>\p{CE=N}</code> , <code>\p{CE}</code>)	(1_114_031)
	<code>\p{Composition_Exclusion: Y*}</code>	(Short: <code>\p{CE=Y}</code> , <code>\p{CE}</code>)	(81)
	<code>\p{Connector_Punctuation}</code>	<code>\p{General_Category=Connector_Punctuation}</code>	(Short: <code>\p{Pc}</code>) (10)
	<code>\p{Control}</code>	<code>\p{Cntrl}</code> (= <code>\p{General_Category=Control}</code>)	(65)
X	<code>\p{Control_Pictures}</code>	<code>\p{Block=Control_Pictures}</code>	(64)
	<code>\p{Copt}</code>	<code>\p{Coptic}</code> (= <code>\p{Script=Coptic}</code>)	(NOT <code>\p{Block=Coptic}</code>) (135)
	<code>\p{Coptic}</code>	<code>\p{Script=Coptic}</code>	(Short: <code>\p{Copt}</code> ; NOT <code>\p{Block=Coptic}</code>) (135)
X	<code>\p{Counting_Rod_Numerals}</code>	<code>\p{Block=Counting_Rod_Numerals}</code>	(32)
	<code>\p{Cprt}</code>	<code>\p{Cypriot}</code> (= <code>\p{Script=Cypriot}</code>)	(55)
	<code>\p{Cs}</code>	<code>\p{Surrogate}</code> (= <code>\p{General_Category=Surrogate}</code>)	(2048)
	<code>\p{Cuneiform}</code>	<code>\p{Script=Cuneiform}</code>	(Short: <code>\p{Xsux}</code> ; NOT <code>\p{Block=Cuneiform}</code>) (982)
X	<code>\p{Cuneiform_Numbers_And_Punctuation}</code>	<code>\p{Block=Cuneiform_Numbers_And_Punctuation}</code>	(128)
	<code>\p{Currency_Symbol}</code>	<code>\p{General_Category=Currency_Symbol}</code>	(Short: <code>\p{Sc}</code>) (47)
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>)	(1102)
	<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>)	(2128)
	<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>)	(9792)
	<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>) (1038)
<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>) (1094)
<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>) (1121)
<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>) (408)
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_Supplement}</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>) (408)
<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>) (420)
<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>) (240)
<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> (240)
<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) (3510)
<code>\p{Decomposition_Type: Non_Canonical}</code>	Union of all non-canonical decompositions (Short: <code>\p{Dt=NonCanon}</code>) (Perl extension) (3510)
<code>\p{Decomposition_Type: None}</code>	(Short: <code>\p{Dt=None}</code>) (1_097_381)
<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>	(284)
<code>\p{Decomposition_Type: Square}</code>	(Short: <code>\p{Dt=Sqr}</code>)	(284)
<code>\p{Decomposition_Type: Sub}</code>	(Short: <code>\p{Dt=Sub}</code>)	(38)
<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>) (111)
<code>\p{Dep: *}</code>	<code>\p{Deprecated: *}</code>	
<code>\p{Deprecated}</code>	<code>\p{Deprecated=Y}</code>	(Short: <code>\p{Dep}</code>) (111)
<code>\p{Deprecated: N*}</code>	(Short: <code>\p{Dep=N}</code> , <code>\p{Dep}</code>)	(1_114_001)
<code>\p{Deprecated: Y*}</code>	(Short: <code>\p{Dep=Y}</code> , <code>\p{Dep}</code>)	(111)
<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>) (150)
<code>\p{Devanagari}</code>	<code>\p{Script=Devanagari}</code>	(Short: <code>\p{Deva}</code> ; NOT <code>\p{Block=Devanagari}</code>) (150)
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>	[0-9] + all other decimal digits (Short: <code>\p{Nd}</code>) (420)
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{Script=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>	
<code>\p{East_Asian_Width: A}</code>	<code>\p{East_Asian_Width=Ambiguous}</code>	(138_746)
<code>\p{East_Asian_Width: Ambiguous}</code>	(Short: <code>\p{Ea=A}</code>)	(138_746)
<code>\p{East_Asian_Width: F}</code>	<code>\p{East_Asian_Width=Fullwidth}</code>	(104)
<code>\p{East_Asian_Width: Fullwidth}</code>	(Short: <code>\p{Ea=F}</code>)	(104)
<code>\p{East_Asian_Width: H}</code>	<code>\p{East_Asian_Width=Halfwidth}</code>	(123)
<code>\p{East_Asian_Width: Halfwidth}</code>	(Short: <code>\p{Ea=H}</code>)	(123)
<code>\p{East_Asian_Width: N}</code>	<code>\p{East_Asian_Width=Neutral}</code>	(801_811)
<code>\p{East_Asian_Width: Na}</code>	<code>\p{East_Asian_Width=Narrow}</code>	(111)
<code>\p{East_Asian_Width: Narrow}</code>	(Short: <code>\p{Ea=Na}</code>)	(111)
<code>\p{East_Asian_Width: Neutral}</code>	(Short: <code>\p{Ea=N}</code>)	(801_811)
<code>\p{East_Asian_Width: W}</code>	<code>\p{East_Asian_Width=Wide}</code>	(173_217)
<code>\p{East_Asian_Width: Wide}</code>	(Short: <code>\p{Ea=W}</code>)	(173_217)
<code>\p{Egyp}</code>	<code>\p{Egyptian_Hieroglyphs}</code>	(= <code>\p{Script=</code>

```

        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{Emoticons} \p{Block=Emoticons} (80)
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}) (12)
    \p{Ethi} \p{Ethiopic} (= \p{Script=Ethiopic}) (NOT
        \p{Block=Ethiopic}) (495)
    \p{Ethiopic} \p{Script=Ethiopic} (Short: \p{Ethi}; NOT
        \p{Block=Ethiopic}) (495)
X \p{Ethiopic_Extended} \p{Block=Ethiopic_Extended} (96)
X \p{Ethiopic_Extended_A} \p{Block=Ethiopic_Extended_A} (48)
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_004_868)
    \p{General_Category: Cased_Letter} [\p{Ll}\p{Lu}\p{Lt}] (Short:
        \p{Gc=LC}, \p{LC}) (3226)
    \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} (865_147)
    \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})
        (47)
    \p{General_Category: Dash_Punctuation} (Short: \p{Gc=Pd}, \p{Pd})

```

```

(21)
\p{General_Category: Decimal_Number} (Short: \p{Gc=Nd}, \p{Nd})
(420)
\p{General_Category: Digit} \p{General_Category=Decimal_Number}
(420)
\p{General_Category: Enclosing_Mark} (Short: \p{Gc=Me}, \p{Me})
(12)
\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} (100_520)
X \p{General_Category: L&} \p{General_Category=Cased_Letter} (3226)
X \p{General_Category: L_} \p{General_Category=Cased_Letter} (3226)
\p{General_Category: LC} \p{General_Category=Cased_Letter} (3226)
\p{General_Category: Letter} (Short: \p{Gc=L}, \p{L}) (100_520)
\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}
(/i= General_Category=Cased_Letter)
(1759)
\p{General_Category: Lm} \p{General_Category=Modifier_Letter} (210)
\p{General_Category: Lo} \p{General_Category=Other_Letter} (97_084)
\p{General_Category: Lowercase_Letter} (Short: \p{Gc=Ll}, \p{Ll};
/i= General_Category=Cased_Letter) (1759)
\p{General_Category: Lt} \p{General_Category=Titlecase_Letter}
(/i= General_Category=Cased_Letter) (31)
\p{General_Category: Lu} \p{General_Category=Uppercase_Letter}
(/i= General_Category=Cased_Letter)
(1436)
\p{General_Category: M} \p{General_Category=Mark} (1498)
\p{General_Category: Mark} (Short: \p{Gc=M}, \p{M}) (1498)
\p{General_Category: Math_Symbol} (Short: \p{Gc=Sm}, \p{Sm}) (948)
\p{General_Category: Mc} \p{General_Category=Spacing_Mark} (287)
\p{General_Category: Me} \p{General_Category=Enclosing_Mark} (12)
\p{General_Category: Mn} \p{General_Category=Nonspacing_Mark}
(1199)
\p{General_Category: Modifier_Letter} (Short: \p{Gc=Lm}, \p{Lm})
(210)
\p{General_Category: Modifier_Symbol} (Short: \p{Gc=Sk}, \p{Sk})
(115)
\p{General_Category: N} \p{General_Category=Number} (1100)
\p{General_Category: Nd} \p{General_Category=Decimal_Number} (420)
\p{General_Category: Nl} \p{General_Category=Letter_Number} (224)
\p{General_Category: No} \p{General_Category=Other_Number} (456)
\p{General_Category: Nonspacing_Mark} (Short: \p{Gc=Mn}, \p{Mn})
(1199)
\p{General_Category: Number} (Short: \p{Gc=N}, \p{N}) (1100)
\p{General_Category: Open_Punctuation} (Short: \p{Gc=Ps}, \p{Ps})
(72)
\p{General_Category: Other} (Short: \p{Gc=C}, \p{C}) (1_004_868)
\p{General_Category: Other_Letter} (Short: \p{Gc=Lo}, \p{Lo})
(97_084)
\p{General_Category: Other_Number} (Short: \p{Gc=No}, \p{No}) (456)

```

```

\p{General_Category: Other_Punctuation} (Short: \p{Gc=Po}, \p{Po})
    (402)
\p{General_Category: Other_Symbol} (Short: \p{Gc=So}, \p{So})
    (4398)
\p{General_Category: P} \p{General_Category=Punctuation} (598)
\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}
    (402)
\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} (598)
\p{General_Category: Punctuation} (Short: \p{Gc=P}, \p{P}) (598)
\p{General_Category: S} \p{General_Category=Symbol} (5508)
\p{General_Category: Sc} \p{General_Category=Currency_Symbol} (47)
\p{General_Category: Separator} (Short: \p{Gc=Z}, \p{Z}) (20)
\p{General_Category: Sk} \p{General_Category=Modifier_Symbol} (115)
\p{General_Category: Sm} \p{General_Category=Math_Symbol} (948)
\p{General_Category: So} \p{General_Category=Other_Symbol} (4398)
\p{General_Category: Space_Separator} (Short: \p{Gc=Zs}, \p{Zs})
    (18)
\p{General_Category: Spacing_Mark} (Short: \p{Gc=Mc}, \p{Mc}) (287)
\p{General_Category: Surrogate} (Short: \p{Gc=Cs}, \p{Cs}) (2048)
\p{General_Category: Symbol} (Short: \p{Gc=S}, \p{S}) (5508)
\p{General_Category: Titlecase_Letter} (Short: \p{Gc=Lt}, \p{Lt};
    /i= General_Category=Cased_Letter) (31)
\p{General_Category: Unassigned} (Short: \p{Gc=Cn}, \p{Cn})
    (865_147)
\p{General_Category: Uppercase_Letter} (Short: \p{Gc=Lu}, \p{Lu};
    /i= General_Category=Cased_Letter) (1436)
\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)
\p{Geor} \p{Georgian} (= \p{Script=Georgian}) (NOT
    \p{Block=Georgian}) (120)
\p{Georgian} \p{Script=Georgian} (Short: \p{Geor}; NOT
    \p{Block=Georgian}) (120)
X \p{Georgian_Supplement} \p{Block=Georgian_Supplement} (48)
\p{Glag} \p{Glagolitic} (= \p{Script=Glagolitic})
    (NOT \p{Block=Glagolitic}) (94)
\p{Glagolitic} \p{Script=Glagolitic} (Short: \p{Glag};
    NOT \p{Block=Glagolitic}) (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>) (108_010)
<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>) (1234)
<code>\p{Gr_Ext: *}</code>	<code>\p{Grapheme_Extend: *}</code>
<code>\p{Graph}</code>	Characters that are graphical (246_832)
<code>\p{Grapheme_Base}</code>	<code>\p{Grapheme_Base=Y}</code> (Short: <code>\p{GrBase}</code>) (108_010)
<code>\p{Grapheme_Base: N*}</code>	(Short: <code>\p{GrBase=N}</code> , <code>\p{GrBase}</code>) (1_006_102)
<code>\p{Grapheme_Base: Y*}</code>	(Short: <code>\p{GrBase=Y}</code> , <code>\p{GrBase}</code>) (108_010)
<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> (1234)
<code>\p{Grapheme_Cluster_Break: Extend}</code>	(Short: <code>\p{GCB=EX}</code>) (1234)
<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_854)
<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> (275)
<code>\p{Grapheme_Cluster_Break: SpacingMark}</code>	(Short: <code>\p{GCB=SM}</code>) (275)
<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_854)
<code>\p{Grapheme_Extend}</code>	<code>\p{Grapheme_Extend=Y}</code> (Short: <code>\p{GrExt}</code>) (1234)
<code>\p{Grapheme_Extend: N*}</code>	(Short: <code>\p{GrExt=N}</code> , <code>\p{GrExt}</code>) (1_112_878)
<code>\p{Grapheme_Extend: Y*}</code>	(Short: <code>\p{GrExt=Y}</code> , <code>\p{GrExt}</code>) (1234)
<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_960)
	<code>\p{Hang}</code>	<code>\p{Hangul}</code> (= <code>\p{Script=Hangul}</code>) (11_739)
	<code>\p{Hangul}</code>	<code>\p{Script=Hangul}</code> (Short: <code>\p{Hang}</code>) (11_739)
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_960)
	<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>) (91)
	<code>\p{Hiragana}</code>	<code>\p{Script=Hiragana}</code> (Short: <code>\p{Hira}</code> ; NOT <code>\p{Block=Hiragana}</code>) (91)
	<code>\p{HorizSpace}</code>	<code>\p{Blank}</code> (19)
	<code>\p{Hst: *}</code>	<code>\p{Hangul_Syllable_Type: *}</code>

D	<code>\p{Hyphen}</code>	<code>\p{Hyphen=Y}</code> (11)
D	<code>\p{Hyphen: N*}</code>	Supplanted by Line_Break property values; see www.unicode.org/reports/tr14 (Single: <code>\P{Hyphen}</code>) (1_114_101)
D	<code>\p{Hyphen: Y*}</code>	Supplanted by Line_Break property values; 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>) (102_675)
	<code>\p{ID_Continue: N*}</code>	(Short: <code>\p{IDC=N}</code> , <code>\P{IDC}</code>) (1_011_437)
	<code>\p{ID_Continue: Y*}</code>	(Short: <code>\p{IDC=Y}</code> , <code>\P{IDC}</code>) (102_675)
	<code>\p{ID_Start}</code>	<code>\p{ID_Start=Y}</code> (Short: <code>\p{IDS}</code>) (100_747)
	<code>\p{ID_Start: N*}</code>	(Short: <code>\p{IDS=N}</code> , <code>\P{IDS}</code>) (1_013_365)
	<code>\p{ID_Start: Y*}</code>	(Short: <code>\p{IDS=Y}</code> , <code>\P{IDS}</code>) (100_747)
	<code>\p{IDC}</code>	<code>\p{ID_Continue}</code> (= <code>\p{ID_Continue=Y}</code>) (102_675)
	<code>\p{IDC: *}</code>	<code>\p{ID_Continue: *}</code>
	<code>\p{Ideo}</code>	<code>\p{Ideographic}</code> (= <code>\p{Ideographic=Y}</code>) (75_630)
	<code>\p{Ideo: *}</code>	<code>\p{Ideographic: *}</code>
	<code>\p{Ideographic}</code>	<code>\p{Ideographic=Y}</code> (Short: <code>\p{Ideo}</code>) (75_630)
	<code>\p{Ideographic: N*}</code>	(Short: <code>\p{Ideo=N}</code> , <code>\P{Ideo}</code>) (1_038_482)
	<code>\p{Ideographic: Y*}</code>	(Short: <code>\p{Ideo=Y}</code> , <code>\P{Ideo}</code>) (75_630)
X	<code>\p{Ideographic_Description_Chacters}</code>	<code>\p{Block=</code> <code>Ideographic_Description_Chacters}</code> (16)
	<code>\p{IDS}</code>	<code>\p{ID_Start}</code> (= <code>\p{ID_Start=Y}</code>) (100_747)
	<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=</code> <code>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=</code> <code>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{Is_*}</code>	<code>\p{Block=IPA_Extensions}</code> (96) <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>	<code>\p{Joining_Group=Teh_Marbuta_Goal}</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)
	<code>\p{Joining_Group: Knotted_Heh}</code>	(Short: <code>\p{Jg=KnottedHeh}</code>) (2)
	<code>\p{Joining_Group: Lam}</code>	(Short: <code>\p{Jg=Lam}</code>) (6)
	<code>\p{Joining_Group: Lamadh}</code>	(Short: <code>\p{Jg=Lamadh}</code>) (1)
	<code>\p{Joining_Group: Meem}</code>	(Short: <code>\p{Jg=Meem}</code>) (3)
	<code>\p{Joining_Group: Mim}</code>	(Short: <code>\p{Jg=Mim}</code>) (1)
	<code>\p{Joining_Group: No_Joining_Group}</code>	(Short: <code>\p{Jg=NoJoiningGroup}</code>) (1_113_882)
	<code>\p{Joining_Group: Noon}</code>	(Short: <code>\p{Jg=Noon}</code>) (8)
	<code>\p{Joining_Group: Nun}</code>	(Short: <code>\p{Jg=Nun}</code>) (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: Teh_Marbuta_Goal} (Short: \p{Jg=TehMarbutaGoal})
    (1)
\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}) (8)
\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} (189)
\p{Joining_Type: Dual_Joining} (Short: \p{Jt=D}) (189)
\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_502)
\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} (1344)
\p{Joining_Type: Transparent} (Short: \p{Jt=T}) (1344)
\p{Joining_Type: U} \p{Joining_Type=Non_Joining} (1_112_502)
\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}) (300)
X \p{Kana_Supplement} \p{Block=Kana_Supplement} (256)
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}) (86)
\p{Katakana} \p{Script=Katakana} (Short: \p{Kana}; NOT
    \p{Block=Katakana}) (300)
X \p{Katakana_Phonetic_Extensions} \p{Block=
    Katakana_Phonetic_Extensions} (16)
\p{Kayah_Li} \p{Script=Kayah_Li} (Short: \p{Kali}) (48)
\p{Khar} \p{Kharoshthi} (= \p{Script=Kharoshthi})

```

	(NOT \p{Block=Kharoshthi}) (65)
\p{Kharoshthi}	\p{Script=Kharoshthi} (Short: \p{Khar}; NOT \p{Block=Kharoshthi}) (65)
\p{Khmer}	\p{Script=Khmer} (Short: \p{Khmr}; NOT \p{Block=Khmer}) (146)
X \p{Khmer_Symbols}	\p{Block=Khmer_Symbols} (32)
\p{Khmr}	\p{Khmer} (= \p{Script=Khmer}) (NOT \p{Block=Khmer}) (146)
\p{Knda}	\p{Kannada} (= \p{Script=Kannada}) (NOT \p{Block=Kannada}) (86)
\p{Kthi}	\p{Kaithi} (= \p{Script=Kaithi}) (NOT \p{Block=Kaithi}) (66)
\p{L}	\p{Letter} (= \p{General_Category=Letter}) (100_520)
\p{L&}	\p{Cased_Letter} (= \p{General_Category=Cased_Letter}) (3226)
\p{L_}	\p{Cased_Letter} (= \p{General_Category=Cased_Letter}) (3226)
\p{Lana}	\p{Tai_Tham} (= \p{Script=Tai_Tham}) (NOT \p{Block=Tai_Tham}) (127)
\p{Lao}	\p{Script=Lao} (NOT \p{Block=Lao}) (65)
\p{Laoo}	\p{Lao} (= \p{Script=Lao}) (NOT \p{Block=Lao}) (65)
\p{Latin}	\p{Script=Latin} (Short: \p{Latn}) (1267)
X \p{Latin_1}	\p{Latin_1_Supplement} (= \p{Block=Latin_1_Supplement}) (128)
X \p{Latin_1_Supplement}	\p{Block=Latin_1_Supplement} (Short: \p{InLatin1}) (128)
X \p{Latin_Extended_A}	\p{Block=Latin_Extended_A} (128)
X \p{Latin_Extended_Additional}	\p{Block=Latin_Extended_Additional} (256)
X \p{Latin_Extended_B}	\p{Block=Latin_Extended_B} (208)
X \p{Latin_Extended_C}	\p{Block=Latin_Extended_C} (32)
X \p{Latin_Extended_D}	\p{Block=Latin_Extended_D} (224)
\p{Latn}	\p{Latin} (= \p{Script=Latin}) (1267)
\p{Lb: *}	\p{Line_Break: *}
\p{LC}	\p{Cased_Letter} (= \p{General_Category=Cased_Letter}) (3226)
\p{Lepc}	\p{Lepcha} (= \p{Script=Lepcha}) (NOT \p{Block=Lepcha}) (74)
\p{Lepcha}	\p{Script=Lepcha} (Short: \p{Lepc}; NOT \p{Block=Lepcha}) (74)
\p{Letter}	\p{General_Category=Letter} (Short: \p{L}) (100_520)
\p{Letter_Number}	\p{General_Category=Letter_Number} (Short: \p{Nl}) (224)
X \p{Letterlike_Symbols}	\p{Block=Letterlike_Symbols} (80)
\p{Limb}	\p{Limbu} (= \p{Script=Limbu}) (NOT \p{Block=Limbu}) (66)
\p{Limbu}	\p{Script=Limbu} (Short: \p{Limb}; NOT \p{Block=Limbu}) (66)
\p{Linb}	\p{Linear_B} (= \p{Script=Linear_B}) (211)
\p{Line_Break: AI}	\p{Line_Break=Ambiguous} (724)
\p{Line_Break: AL}	\p{Line_Break=Alphabetic} (15_797)
\p{Line_Break: Alphabetic}	(Short: \p{Lb=AL}) (15_797)
\p{Line_Break: Ambiguous}	(Short: \p{Lb=AI}) (724)

```

\p{Line_Break: B2}          \p{Line_Break=Break_Both} (1)
\p{Line_Break: BA}          \p{Line_Break=Break_After} (140)
\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}) (140)
\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} (1483)
\p{Line_Break: Combining_Mark} (Short: \p{Lb=CM}) (1483)
\p{Line_Break: Complex_Context} (Short: \p{Lb=SA}) (663)
\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} (18)
\p{Line_Break: Glue} (Short: \p{Lb=GL}) (18)
\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_793)
\p{Line_Break: Ideographic} (Short: \p{Lb=ID}) (161_793)
\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: Inseperable} \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} (412)
\p{Line_Break: Numeric} (Short: \p{Lb=NU}) (412)
\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)
\p{Line_Break: PR}          \p{Line_Break=Prefix_Numeric} (44)
\p{Line_Break: Prefix_Numeric} (Short: \p{Lb=PR}) (44)
\p{Line_Break: QU}          \p{Line_Break=Quotation} (34)
\p{Line_Break: Quotation} (Short: \p{Lb=QU}) (34)
\p{Line_Break: SA}          \p{Line_Break=Complex_Context} (663)
D \p{Line_Break: SG}          \p{Line_Break=Surrogate} (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>) (919_067)
	<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> (919_067)
	<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>) (/i= <code>General_Category=Cased_Letter</code>) (1759)
	<code>\p{Lm}</code>	<code>\p{Modifier_Letter}</code> (= <code>\p{General_Category=Modifier_Letter}</code>) (210)
	<code>\p{Lo}</code>	<code>\p{Other_Letter}</code> (= <code>\p{General_Category=Other_Letter}</code>) (97_084)
	<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> (/i= <code>Cased=Yes</code>) (1918)
	<code>\p{Lower: *}</code>	<code>\p{Lowercase: *}</code>
	<code>\p{Lowercase}</code>	<code>\p{Lower}</code> (= <code>\p{Lowercase=Y}</code>) (/i= <code>Cased=Yes</code>) (1918)
	<code>\p{Lowercase: N*}</code>	(Short: <code>\p{Lower=N}</code> , <code>\p{Lower}</code> ; /i= <code>Cased=No</code>) (1_112_194)
	<code>\p{Lowercase: Y*}</code>	(Short: <code>\p{Lower=Y}</code> , <code>\p{Lower}</code> ; /i= <code>Cased=Yes</code>) (1918)
	<code>\p{Lowercase_Letter}</code>	<code>\p{General_Category=Lowercase_Letter}</code> (Short: <code>\p{Ll}</code> ; /i= <code>General_Category=Cased_Letter</code>) (1759)
	<code>\p{Lt}</code>	<code>\p{Titlecase_Letter}</code> (= <code>\p{General_Category=Titlecase_Letter}</code>) (/i= <code>General_Category=Cased_Letter</code>) (31)
	<code>\p{Lu}</code>	<code>\p{Uppercase_Letter}</code> (= <code>\p{General_Category=Uppercase_Letter}</code>) (/i= <code>General_Category=Cased_Letter</code>) (1436)
	<code>\p{Lyci}</code>	<code>\p{Lycian}</code> (= <code>\p{Script=Lycian}</code>) (NOT

	<code>\p{Lycian}</code>	<code>\p{Block=Lycian}</code>) (29)
	<code>\p{Lydi}</code>	<code>\p{Script=Lycian}</code> (Short: <code>\p{Lyci}</code>); NOT <code>\p{Block=Lycian}</code>) (29)
	<code>\p{Lydian}</code>	<code>\p{Lydian}</code> (= <code>\p{Script=Lydian}</code>) (NOT <code>\p{Block=Lydian}</code>) (27)
	<code>\p{M}</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>) (1498)
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>) (98)
	<code>\p{Mand}</code>	<code>\p{Mandaic}</code> (= <code>\p{Script=Mandaic}</code>) (NOT <code>\p{Block=Mandaic}</code>) (29)
	<code>\p{Mandaic}</code>	<code>\p{Script=Mandaic}</code> (Short: <code>\p{Mand}</code>); NOT <code>\p{Block=Mandaic}</code>) (29)
	<code>\p{Mark}</code>	<code>\p{General_Category=Mark}</code> (Short: <code>\p{M}</code>) (1498)
	<code>\p{Math}</code>	<code>\p{Math=Y}</code> (2165)
	<code>\p{Math: N*}</code>	(Single: <code>\p{Math}</code>) (1_111_947)
	<code>\p{Math: Y*}</code>	(Single: <code>\p{Math}</code>) (2165)
	<code>\p{Math_Symbol}</code>	<code>\p{General_Category=Math_Symbol}</code> (Short: <code>\p{Sm}</code>) (948)
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>) (287)
	<code>\p{Me}</code>	<code>\p{Enclosing_Mark}</code> (= <code>\p{General_Category=Enclosing_Mark}</code>) (12)
	<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_Symbols_And_Pictographs}</code>	<code>\p{Block=Miscellaneous_Symbols_And_Pictographs}</code> (768)
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>) (98)
	<code>\p{Mn}</code>	<code>\p{Nonspacing_Mark}</code> (= <code>\p{General_Category=Nonspacing_Mark}</code>) (1199)
	<code>\p{Modifier_Letter}</code>	<code>\p{General_Category=Modifier_Letter}</code> (Short: <code>\p{Lm}</code>) (210)
	<code>\p{Modifier_Symbol}</code>	<code>\p{General_Category=Modifier_Symbol}</code> (Short: <code>\p{Sk}</code>) (115)
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 \p{Block=Mongolian}) (153)
\p{Mongolian}	\p{Script=Mongolian} (Short: \p{Mong}; NOT \p{Block=Mongolian}) (153)
\p{Mtei}	\p{Meetei_Mayek} (= \p{Script=Meetei_Mayek}) (NOT \p{Block=Meetei_Mayek}) (56)
X \p{Musical_Symbols}	\p{Block=Musical_Symbols} (256)
\p{Myanmar}	\p{Script=Myanmar} (Short: \p{Mymr}; NOT \p{Block=Myanmar}) (188)
X \p{Myanmar_Extended_A}	\p{Block=Myanmar_Extended_A} (32)
\p{Mymr}	\p{Myanmar} (= \p{Script=Myanmar}) (NOT \p{Block=Myanmar}) (188)
\p{N}	\p{Number} (= \p{General_Category=Number}) (1100)
\p{NChar}	\p{Noncharacter_Code_Point} (= \p{Noncharacter_Code_Point=Y}) (66)
\p{NChar: *}	\p{Noncharacter_Code_Point: *}
\p{Nd}	\p{Digit} (= \p{General_Category=Decimal_Number}) (420)
\p{New_Tai_Lue}	\p{Script=New_Tai_Lue} (Short: \p{Talu}; NOT \p{Block=New_Tai_Lue}) (83)
\p{NFC_QC: *}	\p{NFC_Quick_Check: *}
\p{NFC_Quick_Check: M}	\p{NFC_Quick_Check=Maybe} (103)
\p{NFC_Quick_Check: Maybe}	(Short: \p{NFCQC=M}) (103)
\p{NFC_Quick_Check: N}	\p{NFC_Quick_Check=No} (NOT \p{NFC_Quick_Check} NOR \p{NFC_QC} NOR \p{Is_NFC_Quick_Check} NOR \p{Is_NFC_QC}) (1118)
\p{NFC_Quick_Check: No}	(Short: \p{NFCQC=N}; NOT \p{NFC_Quick_Check} NOR \p{NFC_QC} NOR \p{Is_NFC_Quick_Check} NOR \p{Is_NFC_QC}) (1118)
\p{NFC_Quick_Check: Y}	\p{NFC_Quick_Check=Yes} (NOT \p{NFC_Quick_Check} NOR \p{NFC_QC} NOR \p{Is_NFC_Quick_Check} NOR \p{Is_NFC_QC}) (1_112_891)
\p{NFC_Quick_Check: Yes}	(Short: \p{NFCQC=Y}; NOT \p{NFC_Quick_Check} NOR \p{NFC_QC} NOR \p{Is_NFC_Quick_Check} NOR \p{Is_NFC_QC}) (1_112_891)
\p{NFD_QC: *}	\p{NFD_Quick_Check: *}
\p{NFD_Quick_Check: N}	\p{NFD_Quick_Check=No} (NOT \p{NFD_Quick_Check} NOR \p{NFD_QC} NOR \p{Is_NFD_Quick_Check} NOR \p{Is_NFD_QC}) (13_221)
\p{NFD_Quick_Check: No}	(Short: \p{NFDQC=N}; NOT \p{NFD_Quick_Check} NOR \p{NFD_QC} NOR \p{Is_NFD_Quick_Check} NOR \p{Is_NFD_QC}) (13_221)
\p{NFD_Quick_Check: Y}	\p{NFD_Quick_Check=Yes} (NOT \p{NFD_Quick_Check} NOR \p{NFD_QC} NOR \p{Is_NFD_Quick_Check} NOR \p{Is_NFD_QC}) (1_100_891)
\p{NFD_Quick_Check: Yes}	(Short: \p{NFDQC=Y}; NOT \p{NFD_Quick_Check} NOR \p{NFD_QC} NOR \p{Is_NFD_Quick_Check} NOR

```

        \p{Is_NFD_QC}) (1_100_891)
\p{NFKC_QC: *}          \p{NFKC_Quick_Check: *}
\p{NFKC_Quick_Check: M} \p{NFKC_Quick_Check=Maybe} (103)
\p{NFKC_Quick_Check: Maybe} (Short: \p{NFKCQC=M}) (103)
\p{NFKC_Quick_Check: N} \p{NFKC_Quick_Check=No} (NOT
        \P{NFKC_Quick_Check} NOR \P{NFKC_QC} NOR
        \P{Is_NFKC_Quick_Check} NOR
        \P{Is_NFKC_QC}) (4640)
\p{NFKC_Quick_Check: No} (Short: \p{NFKCQC=N}; NOT
        \P{NFKC_Quick_Check} NOR \P{NFKC_QC} NOR
        \P{Is_NFKC_Quick_Check} NOR
        \P{Is_NFKC_QC}) (4640)
\p{NFKC_Quick_Check: Y} \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_369)
\p{NFKC_Quick_Check: Yes} (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_369)
\p{NFKD_QC: *}          \p{NFKD_Quick_Check: *}
\p{NFKD_Quick_Check: N} \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_731)
\p{NFKD_Quick_Check: No} (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_731)
\p{NFKD_Quick_Check: Y} \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_381)
\p{NFKD_Quick_Check: Yes} (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_381)
\p{Nko}                 \p{Script=Nko} (NOT \p{Nko}) (59)
\p{Nkoo}                \p{Nko} (= \p{Script=Nko}) (NOT \p{Nko})
        (59)
\p{Nl}                  \p{Letter_Number} (= \p{General_Category=
        Letter_Number}) (224)
\p{No}                  \p{Other_Number} (= \p{General_Category=
        Other_Number}) (456)
X \p{No_Block}          \p{Block=No_Block} (861_664)
\p{Noncharacter_Code_Point} \p{Noncharacter_Code_Point=Y} (Short:
        \p{NChar}) (66)
\p{Noncharacter_Code_Point: N*} (Short: \p{NChar=N}, \P{NChar})
        (1_114_046)
\p{Noncharacter_Code_Point: Y*} (Short: \p{NChar=Y}, \P{NChar})
        (66)
\p{Nonspacing_Mark}    \p{General_Category=Nonspacing_Mark}
        (Short: \p{Mn}) (1199)
\p{Nt: *}              \p{Numeric_Type: *}
\p{Number}              \p{General_Category=Number} (Short: \p{N})
        (1100)

```

X	$\backslash p\{Number_Forms\}$	$\backslash p\{Block=Number_Forms\}$	(64)
	$\backslash p\{Numeric_Type: De\}$	$\backslash p\{Numeric_Type=Decimal\}$	(420)
	$\backslash p\{Numeric_Type: Decimal\}$	(Short: $\backslash p\{Nt=De\}$)	(420)
	$\backslash p\{Numeric_Type: Di\}$	$\backslash p\{Numeric_Type=Digit\}$	(128)
	$\backslash p\{Numeric_Type: Digit\}$	(Short: $\backslash p\{Nt=Di\}$)	(128)
	$\backslash p\{Numeric_Type: None\}$	(Short: $\backslash p\{Nt=None\}$)	(1_112_935)
	$\backslash p\{Numeric_Type: Nu\}$	$\backslash p\{Numeric_Type=Numeric\}$	(629)
	$\backslash p\{Numeric_Type: Numeric\}$	(Short: $\backslash p\{Nt=Nu\}$)	(629)
T	$\backslash p\{Numeric_Value: -1/2\}$	(Short: $\backslash p\{Nv=-1/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 0\}$	(Short: $\backslash p\{Nv=0\}$)	(56)
T	$\backslash p\{Numeric_Value: 1/16\}$	(Short: $\backslash p\{Nv=1/16\}$)	(3)
T	$\backslash p\{Numeric_Value: 1/10\}$	(Short: $\backslash p\{Nv=1/10\}$)	(1)
T	$\backslash p\{Numeric_Value: 1/9\}$	(Short: $\backslash p\{Nv=1/9\}$)	(1)
T	$\backslash p\{Numeric_Value: 1/8\}$	(Short: $\backslash p\{Nv=1/8\}$)	(5)
T	$\backslash p\{Numeric_Value: 1/7\}$	(Short: $\backslash p\{Nv=1/7\}$)	(1)
T	$\backslash p\{Numeric_Value: 1/6\}$	(Short: $\backslash p\{Nv=1/6\}$)	(2)
T	$\backslash p\{Numeric_Value: 3/16\}$	(Short: $\backslash p\{Nv=3/16\}$)	(3)
T	$\backslash p\{Numeric_Value: 1/5\}$	(Short: $\backslash p\{Nv=1/5\}$)	(1)
T	$\backslash p\{Numeric_Value: 1/4\}$	(Short: $\backslash p\{Nv=1/4\}$)	(9)
T	$\backslash p\{Numeric_Value: 1/3\}$	(Short: $\backslash p\{Nv=1/3\}$)	(4)
T	$\backslash p\{Numeric_Value: 3/8\}$	(Short: $\backslash p\{Nv=3/8\}$)	(1)
T	$\backslash p\{Numeric_Value: 2/5\}$	(Short: $\backslash p\{Nv=2/5\}$)	(1)
T	$\backslash p\{Numeric_Value: 1/2\}$	(Short: $\backslash p\{Nv=1/2\}$)	(10)
T	$\backslash p\{Numeric_Value: 3/5\}$	(Short: $\backslash p\{Nv=3/5\}$)	(1)
T	$\backslash p\{Numeric_Value: 5/8\}$	(Short: $\backslash p\{Nv=5/8\}$)	(1)
T	$\backslash p\{Numeric_Value: 2/3\}$	(Short: $\backslash p\{Nv=2/3\}$)	(5)
T	$\backslash p\{Numeric_Value: 3/4\}$	(Short: $\backslash p\{Nv=3/4\}$)	(6)
T	$\backslash p\{Numeric_Value: 4/5\}$	(Short: $\backslash p\{Nv=4/5\}$)	(1)
T	$\backslash p\{Numeric_Value: 5/6\}$	(Short: $\backslash p\{Nv=5/6\}$)	(2)
T	$\backslash p\{Numeric_Value: 7/8\}$	(Short: $\backslash p\{Nv=7/8\}$)	(1)
T	$\backslash p\{Numeric_Value: 1\}$	(Short: $\backslash p\{Nv=1\}$)	(93)
T	$\backslash p\{Numeric_Value: 3/2\}$	(Short: $\backslash p\{Nv=3/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 2\}$	(Short: $\backslash p\{Nv=2\}$)	(96)
T	$\backslash p\{Numeric_Value: 5/2\}$	(Short: $\backslash p\{Nv=5/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 3\}$	(Short: $\backslash p\{Nv=3\}$)	(98)
T	$\backslash p\{Numeric_Value: 7/2\}$	(Short: $\backslash p\{Nv=7/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 4\}$	(Short: $\backslash p\{Nv=4\}$)	(89)
T	$\backslash p\{Numeric_Value: 9/2\}$	(Short: $\backslash p\{Nv=9/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 5\}$	(Short: $\backslash p\{Nv=5\}$)	(86)
T	$\backslash p\{Numeric_Value: 11/2\}$	(Short: $\backslash p\{Nv=11/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 6\}$	(Short: $\backslash p\{Nv=6\}$)	(78)
T	$\backslash p\{Numeric_Value: 13/2\}$	(Short: $\backslash p\{Nv=13/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 7\}$	(Short: $\backslash p\{Nv=7\}$)	(77)
T	$\backslash p\{Numeric_Value: 15/2\}$	(Short: $\backslash p\{Nv=15/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 8\}$	(Short: $\backslash p\{Nv=8\}$)	(73)
T	$\backslash p\{Numeric_Value: 17/2\}$	(Short: $\backslash p\{Nv=17/2\}$)	(1)
T	$\backslash p\{Numeric_Value: 9\}$	(Short: $\backslash p\{Nv=9\}$)	(77)
T	$\backslash p\{Numeric_Value: 10\}$	(Short: $\backslash p\{Nv=10\}$)	(39)
T	$\backslash p\{Numeric_Value: 11\}$	(Short: $\backslash p\{Nv=11\}$)	(6)
T	$\backslash p\{Numeric_Value: 12\}$	(Short: $\backslash p\{Nv=12\}$)	(6)
T	$\backslash p\{Numeric_Value: 13\}$	(Short: $\backslash p\{Nv=13\}$)	(4)
T	$\backslash p\{Numeric_Value: 14\}$	(Short: $\backslash p\{Nv=14\}$)	(4)
T	$\backslash p\{Numeric_Value: 15\}$	(Short: $\backslash p\{Nv=15\}$)	(4)
T	$\backslash p\{Numeric_Value: 16\}$	(Short: $\backslash p\{Nv=16\}$)	(5)
T	$\backslash p\{Numeric_Value: 17\}$	(Short: $\backslash p\{Nv=17\}$)	(5)
T	$\backslash p\{Numeric_Value: 18\}$	(Short: $\backslash p\{Nv=18\}$)	(5)

T	<code>\p{Numeric_Value: 30000}</code>	(= 3.0e+04) (Short: <code>\p{Nv=30000}</code>) (1)
T	<code>\p{Numeric_Value: 40000}</code>	(= 4.0e+04) (Short: <code>\p{Nv=40000}</code>) (1)
T	<code>\p{Numeric_Value: 50000}</code>	(= 5.0e+04) (Short: <code>\p{Nv=50000}</code>) (4)
T	<code>\p{Numeric_Value: 60000}</code>	(= 6.0e+04) (Short: <code>\p{Nv=60000}</code>) (1)
T	<code>\p{Numeric_Value: 70000}</code>	(= 7.0e+04) (Short: <code>\p{Nv=70000}</code>) (1)
T	<code>\p{Numeric_Value: 80000}</code>	(= 8.0e+04) (Short: <code>\p{Nv=80000}</code>) (1)
T	<code>\p{Numeric_Value: 90000}</code>	(= 9.0e+04) (Short: <code>\p{Nv=90000}</code>) (1)
T	<code>\p{Numeric_Value: 100000}</code>	(= 1.0e+05) (Short: <code>\p{Nv=100000}</code>) (1)
T	<code>\p{Numeric_Value: 100000000}</code>	(= 1.0e+08) (Short: <code>\p{Nv=100000000}</code>) (2)
T	<code>\p{Numeric_Value: 1000000000000}</code>	(= 1.0e+12) (Short: <code>\p{Nv=1000000000000}</code>) (1)
	<code>\p{Numeric_Value: NaN}</code>	(Short: <code>\p{Nv=NaN}</code>) (1_112_935)
	<code>\p{Nv: *}</code>	<code>\p{Numeric_Value: *}</code>
D	<code>\p{OAlpha}</code>	<code>\p{Other_Alphabetic}</code> (= <code>\p{Other_Alphabetic=Y}</code>) (795)
D	<code>\p{OAlpha: *}</code>	<code>\p{Other_Alphabetic: *}</code>
D	<code>\p{ODI}</code>	<code>\p{Other_Default_Ignorable_Code_Point}</code> (= <code>\p{Other_Default_Ignorable_Code_Point=Y}</code>) (3778)
D	<code>\p{ODI: *}</code>	<code>\p{Other_Default_Ignorable_Code_Point: *}</code>
	<code>\p{Ogam}</code>	<code>\p{Ogham}</code> (= <code>\p{Script=Ogham}</code>) (NOT <code>\p{Block=Ogham}</code>) (29)
	<code>\p{Ogham}</code>	<code>\p{Script=Ogham}</code> (Short: <code>\p{Ogam}</code> ; NOT <code>\p{Block=Ogham}</code>) (29)
D	<code>\p{OGr_Ext}</code>	<code>\p{Other_Grapheme_Extend}</code> (= <code>\p{Other_Grapheme_Extend=Y}</code>) (23)
D	<code>\p{OGr_Ext: *}</code>	<code>\p{Other_Grapheme_Extend: *}</code>
D	<code>\p{OIDC}</code>	<code>\p{Other_ID_Continue}</code> (= <code>\p{Other_ID_Continue=Y}</code>) (12)
D	<code>\p{OIDC: *}</code>	<code>\p{Other_ID_Continue: *}</code>
D	<code>\p{OIDS}</code>	<code>\p{Other_ID_Start}</code> (= <code>\p{Other_ID_Start=Y}</code>) (4)
D	<code>\p{OIDS: *}</code>	<code>\p{Other_ID_Start: *}</code>
	<code>\p{Ol_Chiki}</code>	<code>\p{Script=Ol_Chiki}</code> (Short: <code>\p{Olck}</code>) (48)
	<code>\p{Olck}</code>	<code>\p{Ol_Chiki}</code> (= <code>\p{Script=Ol_Chiki}</code>) (48)
	<code>\p{Old_Italic}</code>	<code>\p{Script=Old_Italic}</code> (Short: <code>\p{Ital}</code> ; NOT <code>\p{Block=Old_Italic}</code>) (35)
	<code>\p{Old_Persian}</code>	<code>\p{Script=Old_Persian}</code> (Short: <code>\p{Xpeo}</code> ; NOT <code>\p{Block=Old_Persian}</code>) (50)
	<code>\p{Old_South_Arabian}</code>	<code>\p{Script=Old_South_Arabian}</code> (Short: <code>\p{Sarab}</code>) (32)
	<code>\p{Old_Turkic}</code>	<code>\p{Script=Old_Turkic}</code> (Short: <code>\p{Orkh}</code> ; NOT <code>\p{Block=Old_Turkic}</code>) (73)
D	<code>\p{OLower}</code>	<code>\p{Other_Lowercase}</code> (= <code>\p{Other_Lowercase=Y}</code>) (159)
D	<code>\p{OLower: *}</code>	<code>\p{Other_Lowercase: *}</code>
D	<code>\p{OMath}</code>	<code>\p{Other_Math}</code> (= <code>\p{Other_Math=Y}</code>) (1217)
D	<code>\p{OMath: *}</code>	<code>\p{Other_Math: *}</code>
	<code>\p{Open_Punctuation}</code>	<code>\p{General_Category=Open_Punctuation}</code> (Short: <code>\p{Ps}</code>) (72)
X	<code>\p{Optical_Character_Recognition}</code>	<code>\p{Block=Optical_Character_Recognition}</code> (32)
	<code>\p{Oriya}</code>	<code>\p{Script=Oriya}</code> (Short: <code>\p{Orya}</code> ; NOT <code>\p{Block=Oriya}</code>) (90)
	<code>\p{Orkh}</code>	<code>\p{Old_Turkic}</code> (= <code>\p{Script=Old_Turkic}</code>)

	(NOT \p{Block=Old_Turkic}) (73)
\p{Orya}	\p{Oriya} (= \p{Script=Oriya}) (NOT \p{Block=Oriya}) (90)
\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_004_868)
D \p{Other_Alphabetic}	\p{Other_Alphabetic=Y} (Short: \p{OAlpha}) (795)
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_317)
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}) (795)
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}) (12)
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_100)
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: $\backslash\text{p}\{\text{OIDC}=\text{Y}\}$, $\backslash\text{P}\{\text{OIDC}\}$) (12)
D $\backslash\text{p}\{\text{Other_ID_Start}\}$	$\backslash\text{p}\{\text{Other_ID_Start}=\text{Y}\}$ (Short: $\backslash\text{p}\{\text{OIDS}\}$) (4)
D $\backslash\text{p}\{\text{Other_ID_Start: N}^*\}$	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: $\backslash\text{p}\{\text{Lo}\}$) (97_084)
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: $\backslash\text{p}\{\text{OMath}\}$) (1217)
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_895)
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: $\backslash\text{p}\{\text{OMath}=\text{Y}\}$, $\backslash\text{P}\{\text{OMath}\}$) (1217)
$\backslash\text{p}\{\text{Other_Number}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Number}\}$ (Short: $\backslash\text{p}\{\text{No}\}$) (456)
$\backslash\text{p}\{\text{Other_Punctuation}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Punctuation}\}$ (Short: $\backslash\text{p}\{\text{Po}\}$) (402)
$\backslash\text{p}\{\text{Other_Symbol}\}$	$\backslash\text{p}\{\text{General_Category}=\text{Other_Symbol}\}$ (Short: $\backslash\text{p}\{\text{So}\}$) (4398)
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: <code>\p{OUpper=Y}</code> , <code>\p{OUpper}</code>) (42)
D <code>\p{OUpper}</code>	<code>\p{Other_Uppercase}</code> (= <code>\p{Other_Uppercase=Y}</code>) (42)
D <code>\p{OUpper: *}</code> <code>\p{P}</code>	<code>\p{Other_Uppercase: *}</code> <code>\p{Punct}</code> (= <code>\p{General_Category=Punctuation}</code>) (598)
<code>\p{Paragraph_Separator}</code>	<code>\p{General_Category=Paragraph_Separator}</code> (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{Pat_WS}</code>	<code>\p{Pattern_Syntax: *}</code> <code>\p{Pattern_White_Space}</code> (= <code>\p{Pattern_White_Space=Y}</code>) (11)
<code>\p{Pat_WS: *}</code> <code>\p{Pattern_Syntax}</code>	<code>\p{Pattern_White_Space: *}</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 = [A-Za-z0-9_] (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{Phli}</code>	<code>\p{Block=Phaistos_Disc}</code> (48) <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=</code>

	Initial_Punctuation)) (12)
X <code>\p{Playing_Cards}</code>	<code>\p{Block=Playing_Cards}</code> (96)
<code>\p{Po}</code>	<code>\p{Other_Punctuation}</code> (=
	<code>\p{General_Category=Other_Punctuation}</code>)
	(402)
<code>\p{PosixAlnum}</code>	[A-Za-z0-9] (62)
<code>\p{PosixAlpha}</code>	[A-Za-z] (52)
<code>\p{PosixBlank}</code>	<code>\t</code> and ' ' (2)
<code>\p{PosixCntrl}</code>	ASCII control characters: NUL, SOH, STX,
	ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT,
	FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4,
	NAK, SYN, ETB, CAN, EOM, SUB, ESC, FS,
	GS, RS, US, and DEL (33)
<code>\p{PosixDigit}</code>	[0-9] (10)
<code>\p{PosixGraph}</code>	[-!"#\$%&'()*+,-./:;<>?@[\\]^_`{ }~0-9A-Za-
	z] (94)
<code>\p{PosixLower}</code>	[a-z] (/i= PosixAlpha) (26)
<code>\p{PosixPrint}</code>	[- 0-9A-Za-
	z!"#\$%&'()*+,-./:;<>?@[\\]^_`{ }~] (95)
<code>\p{PosixPunct}</code>	[-!"#\$%&'()*+,-./:;<>?@[\\]^_`{ }~] (32)
<code>\p{PosixSpace}</code>	<code>\t</code> , <code>\n</code> , <code>\cK</code> , <code>\f</code> , <code>\r</code> , and ' '. (<code>\cK</code> is
	vertical tab) (6)
<code>\p{PosixUpper}</code>	[A-Z] (/i= PosixAlpha) (26)
<code>\p{PosixWord}</code>	<code>\p{PerlWord}</code> (63)
<code>\p{PosixXDigit}</code>	[0-9A-Fa-f] (22)
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)
T \p{Present_In: 6.0}	Code point's usage introduced in version 6.0 or earlier (Short: \p{In=6.0}) (Perl extension) (249_031)
\p{Present_In: Unassigned}	\p{Age=Unassigned} (Short: \p{In=Unassigned}) (Perl extension) (865_081)
\p{Print}	Characters that are graphical plus space characters (but no controls) (246_850)
\p{Private_Use}	\p{General_Category=Private_Use} (Short: \p{Co}; NOT \p{Private_Use_Area}) (137_468)
X \p{Private_Use_Area}	\p{Block=Private_Use_Area} (Short: \p{InPrivateUse}) (6400)
\p{Prti}	\p{Inscriptional_Parthian} (= \p{Script=Inscriptional_Parthian}) (NOT \p{Block=Inscriptional_Parthian}) (30)
\p{Ps}	\p{Open_Punctuation} (= \p{General_Category=Open_Punctuation}) (72)
\p{Punct}	\p{General_Category=Punctuation} (Short: \p{P}) (598)
\p{Punctuation}	\p{Punct} (= \p{General_Category=Punctuation}) (598)
\p{Qaac}	\p{Coptic} (= \p{Script=Coptic}) (NOT \p{Block=Coptic}) (135)
\p{Qaai}	\p{Inherited} (= \p{Script=Inherited}) (523)
\p{QMark}	\p{Quotation_Mark} (= \p{Quotation_Mark=Y}) (29)
\p{QMark: *}	\p{Quotation_Mark: *}
\p{Quotation_Mark}	\p{Quotation_Mark=Y} (Short: \p{QMark}) (29)
\p{Quotation_Mark: N*}	(Short: \p{QMark=N}, \p{QMark}) (1_114_083)
\p{Quotation_Mark: Y*}	(Short: \p{QMark=Y}, \p{QMark}) (29)
\p{Radical}	\p{Radical=Y} (329)
\p{Radical: N*}	(Single: \p{Radical}) (1_113_783)
\p{Radical: Y*}	(Single: \p{Radical}) (329)
\p{Rejang}	\p{Script=Rejang} (Short: \p{Rjng}; NOT \p{Block=Rejang}) (37)
\p{Rjng}	\p{Rejang} (= \p{Script=Rejang}) (NOT \p{Block=Rejang}) (37)
X \p{Rumi_Numeral_Symbols}	\p{Block=Rumi_Numeral_Symbols} (32)
\p{Runic}	\p{Script=Runic} (Short: \p{Runr}; NOT \p{Block=Runic}) (78)
\p{Runr}	\p{Runic} (= \p{Script=Runic}) (NOT \p{Block=Runic}) (78)
\p{S}	\p{Symbol} (= \p{General_Category=Symbol}) (5508)
\p{Samaritan}	\p{Script=Samaritan} (Short: \p{Samr}; NOT \p{Block=Samaritan}) (61)
\p{Samr}	\p{Samaritan} (= \p{Script=Samaritan}) (NOT \p{Block=Samaritan}) (61)
\p{Sarab}	\p{Old_South_Arabian} (= \p{Script=Old_South_Arabian}) (32)
\p{Saur}	\p{Saurashtra} (= \p{Script=Saurashtra}) (NOT \p{Block=Saurashtra}) (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>) (47)
<code>\p{Sc: *}</code>	<code>\p{Script: *}</code>
<code>\p{Script: Arab}</code>	<code>\p{Script=Arabic}</code> (1051)
<code>\p{Script: Arabic}</code>	(Short: <code>\p{Sc=Arab}</code> , <code>\p{Arab}</code>) (1051)
<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> (657)
<code>\p{Script: Bamum}</code>	(Short: <code>\p{Sc=Bamu}</code> , <code>\p{Bamu}</code>) (657)
<code>\p{Script: Batak}</code>	(Short: <code>\p{Sc=Batk}</code> , <code>\p{Batk}</code>) (56)
<code>\p{Script: Batk}</code>	<code>\p{Script=Batak}</code> (56)
<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> (70)
<code>\p{Script: Bopomofo}</code>	(Short: <code>\p{Sc=Bopo}</code> , <code>\p{Bopo}</code>) (70)
<code>\p{Script: Brah}</code>	<code>\p{Script=Brahmi}</code> (108)
<code>\p{Script: Brahmi}</code>	(Short: <code>\p{Sc=Brah}</code> , <code>\p{Brah}</code>) (108)
<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>) (6379)
<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>) (408)
<code>\p{Script: Cyrl}</code>	<code>\p{Script=Cyrillic}</code> (408)
<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> (150)
<code>\p{Script: Devanagari}</code>	(Short: <code>\p{Sc=Deva}</code> , <code>\p{Deva}</code>) (150)
<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> (495)

<code>\p{Script: Ethiopic}</code>	(Short: <code>\p{Sc=Ethi}</code> , <code>\p{Ethi}</code>) (495)
<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_960)
<code>\p{Script: Hang}</code>	<code>\p{Script=Hangul}</code> (11_739)
<code>\p{Script: Hangul}</code>	(Short: <code>\p{Sc=Hang}</code> , <code>\p{Hang}</code>) (11_739)
<code>\p{Script: Hani}</code>	<code>\p{Script=Han}</code> (75_960)
<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> (91)
<code>\p{Script: Hiragana}</code>	(Short: <code>\p{Sc=Hira}</code> , <code>\p{Hira}</code>) (91)
<code>\p{Script: Imperial_Aramaic}</code>	(Short: <code>\p{Sc=Armi}</code> , <code>\p{Armi}</code>) (31)
<code>\p{Script: Inherited}</code>	(Short: <code>\p{Sc=Zinh}</code> , <code>\p{Zinh}</code>) (523)
<code>\p{Script: Inscriptional_Pahlavi}</code>	(Short: <code>\p{Sc=Phli}</code> , <code>\p{Phli}</code>) (27)
<code>\p{Script: Inscriptional_Parthian}</code>	(Short: <code>\p{Sc=Prti}</code> , <code>\p{Prti}</code>) (30)
<code>\p{Script: Ital}</code>	<code>\p{Script=Old_Italic}</code> (35)
<code>\p{Script: Java}</code>	<code>\p{Script=Javanese}</code> (91)
<code>\p{Script: Javanese}</code>	(Short: <code>\p{Sc=Java}</code> , <code>\p{Java}</code>) (91)
<code>\p{Script: Kaithi}</code>	(Short: <code>\p{Sc=Kthi}</code> , <code>\p{Kthi}</code>) (66)
<code>\p{Script: Kali}</code>	<code>\p{Script=Kayah_Li}</code> (48)
<code>\p{Script: Kana}</code>	<code>\p{Script=Katakana}</code> (300)
<code>\p{Script: Kannada}</code>	(Short: <code>\p{Sc=Knda}</code> , <code>\p{Knda}</code>) (86)
<code>\p{Script: Katakana}</code>	(Short: <code>\p{Sc=Kana}</code> , <code>\p{Kana}</code>) (300)
<code>\p{Script: Kayah_Li}</code>	(Short: <code>\p{Sc=Kali}</code> , <code>\p{Kali}</code>) (48)
<code>\p{Script: Khar}</code>	<code>\p{Script=Kharoshthi}</code> (65)
<code>\p{Script: Kharoshthi}</code>	(Short: <code>\p{Sc=Khar}</code> , <code>\p{Khar}</code>) (65)
<code>\p{Script: Khmer}</code>	(Short: <code>\p{Sc=Khmr}</code> , <code>\p{Khmr}</code>) (146)
<code>\p{Script: Khmr}</code>	<code>\p{Script=Khmer}</code> (146)
<code>\p{Script: Knda}</code>	<code>\p{Script=Kannada}</code> (86)
<code>\p{Script: Kthi}</code>	<code>\p{Script=Kaithi}</code> (66)
<code>\p{Script: Lana}</code>	<code>\p{Script=Tai_Tham}</code> (127)
<code>\p{Script: Lao}</code>	(Short: <code>\p{Sc=Lao}</code> , <code>\p{Lao}</code>) (65)
<code>\p{Script: Laoo}</code>	<code>\p{Script=Lao}</code> (65)
<code>\p{Script: Latin}</code>	(Short: <code>\p{Sc=Latn}</code> , <code>\p{Latn}</code>) (1267)
<code>\p{Script: Latn}</code>	<code>\p{Script=Latin}</code> (1267)
<code>\p{Script: Lepc}</code>	<code>\p{Script=Lepcha}</code> (74)
<code>\p{Script: Lepcha}</code>	(Short: <code>\p{Sc=Lepc}</code> , <code>\p{Lepc}</code>) (74)
<code>\p{Script: Limb}</code>	<code>\p{Script=Limbu}</code> (66)
<code>\p{Script: Limbu}</code>	(Short: <code>\p{Sc=Limb}</code> , <code>\p{Limb}</code>) (66)
<code>\p{Script: Linb}</code>	<code>\p{Script=Linear_B}</code> (211)
<code>\p{Script: Linear_B}</code>	(Short: <code>\p{Sc=Linb}</code> , <code>\p{Linb}</code>) (211)
<code>\p{Script: Lisu}</code>	(Short: <code>\p{Sc=Lisu}</code> , <code>\p{Lisu}</code>) (48)

<code>\p{Script: Lyci}</code>	<code>\p{Script=Lycian}</code> (29)
<code>\p{Script: Lycian}</code>	(Short: <code>\p{Sc=Lyci}</code> , <code>\p{Lyci}</code>) (29)
<code>\p{Script: Lydi}</code>	<code>\p{Script=Lydian}</code> (27)
<code>\p{Script: Lydian}</code>	(Short: <code>\p{Sc=Lydi}</code> , <code>\p{Lydi}</code>) (27)
<code>\p{Script: Malayalam}</code>	(Short: <code>\p{Sc=Mlym}</code> , <code>\p{Mlym}</code>) (98)
<code>\p{Script: Mand}</code>	<code>\p{Script=Mandaic}</code> (29)
<code>\p{Script: Mandaic}</code>	(Short: <code>\p{Sc=Mand}</code> , <code>\p{Mand}</code>) (29)
<code>\p{Script: Meetei_Mayek}</code>	(Short: <code>\p{Sc=Mtei}</code> , <code>\p{Mtei}</code>) (56)
<code>\p{Script: Mlym}</code>	<code>\p{Script=Malayalam}</code> (98)
<code>\p{Script: Mong}</code>	<code>\p{Script=Mongolian}</code> (153)
<code>\p{Script: Mongolian}</code>	(Short: <code>\p{Sc=Mong}</code> , <code>\p{Mong}</code>) (153)
<code>\p{Script: Mtei}</code>	<code>\p{Script=Meetei_Mayek}</code> (56)
<code>\p{Script: Myanmar}</code>	(Short: <code>\p{Sc=Mymr}</code> , <code>\p{Mymr}</code>) (188)
<code>\p{Script: Mymr}</code>	<code>\p{Script=Myanmar}</code> (188)
<code>\p{Script: New_Tai_Lue}</code>	(Short: <code>\p{Sc=Talu}</code> , <code>\p{Talu}</code>) (83)
<code>\p{Script: Nko}</code>	(Short: <code>\p{Sc=Nko}</code> , <code>\p{Nko}</code>) (59)
<code>\p{Script: Nkoo}</code>	<code>\p{Script=Nko}</code> (59)
<code>\p{Script: Ogam}</code>	<code>\p{Script=Ogham}</code> (29)
<code>\p{Script: Ogham}</code>	(Short: <code>\p{Sc=Ogam}</code> , <code>\p{Ogam}</code>) (29)
<code>\p{Script: Ol_Chiki}</code>	(Short: <code>\p{Sc=Olck}</code> , <code>\p{Olck}</code>) (48)
<code>\p{Script: Olck}</code>	<code>\p{Script=Ol_Chiki}</code> (48)
<code>\p{Script: Old_Italic}</code>	(Short: <code>\p{Sc=Ital}</code> , <code>\p{Ital}</code>) (35)
<code>\p{Script: Old_Persian}</code>	(Short: <code>\p{Sc=Xpeo}</code> , <code>\p{Xpeo}</code>) (50)
<code>\p{Script: Old_South_Arabian}</code>	(Short: <code>\p{Sc=Sarb}</code> , <code>\p{Sarb}</code>) (32)
<code>\p{Script: Old_Turkic}</code>	(Short: <code>\p{Sc=Orkh}</code> , <code>\p{Orkh}</code>) (73)
<code>\p{Script: Oriya}</code>	(Short: <code>\p{Sc=Orya}</code> , <code>\p{Orya}</code>) (90)
<code>\p{Script: Orkh}</code>	<code>\p{Script=Old_Turkic}</code> (73)
<code>\p{Script: Orya}</code>	<code>\p{Script=Oriya}</code> (90)
<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: Shaw}</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>	<code>(Short: \p{Sc=Syrc}, \p{Syrc})</code>	(77)
<code>\p{Script: Tagalog}</code>	<code>(Short: \p{Sc=Tglg}, \p{Tglg})</code>	(20)
<code>\p{Script: Tagb}</code>	<code>\p{Script=Tagbanwa}</code>	(18)
<code>\p{Script: Tagbanwa}</code>	<code>(Short: \p{Sc=Tagb}, \p{Tagb})</code>	(18)
<code>\p{Script: Tai_Le}</code>	<code>(Short: \p{Sc=Tale}, \p{Tale})</code>	(35)
<code>\p{Script: Tai_Tham}</code>	<code>(Short: \p{Sc=Lana}, \p{Lana})</code>	(127)
<code>\p{Script: Tai_Viet}</code>	<code>(Short: \p{Sc=Tavt}, \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>	<code>(Short: \p{Sc=Taml}, \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>	<code>(Short: \p{Sc=Telu}, \p{Telu})</code>	(93)
<code>\p{Script: Tfng}</code>	<code>\p{Script=Tifinagh}</code>	(57)
<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>	<code>(Short: \p{Sc=Thaa}, \p{Thaa})</code>	(50)
<code>\p{Script: Thai}</code>	<code>(Short: \p{Sc=Thai}, \p{Thai})</code>	(86)
<code>\p{Script: Tibetan}</code>	<code>(Short: \p{Sc=Tibt}, \p{Tibt})</code>	(207)
<code>\p{Script: Tibt}</code>	<code>\p{Script=Tibetan}</code>	(207)
<code>\p{Script: Tifinagh}</code>	<code>(Short: \p{Sc=Tfng}, \p{Tfng})</code>	(57)
<code>\p{Script: Ugar}</code>	<code>\p{Script=Ugaritic}</code>	(31)
<code>\p{Script: Ugaritic}</code>	<code>(Short: \p{Sc=Ugar}, \p{Ugar})</code>	(31)
<code>\p{Script: Unknown}</code>	<code>(Short: \p{Sc=Zzzz}, \p{Zzzz})</code>	(1_004_663)
<code>\p{Script: Vai}</code>	<code>(Short: \p{Sc=Vai}, \p{Vai})</code>	(300)
<code>\p{Script: Vaih}</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: Yih}</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>	(6379)
<code>\p{Script: Zzzz}</code>	<code>\p{Script=Unknown}</code>	(1_004_663)
<code>\p{SD}</code>	<code>\p{Soft_Dotted} (= \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>	(1502)
<code>\p{Sentence_Break: Extend}</code>	<code>(Short: \p{SB=EX})</code>	(1502)
<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>	(97_369)
<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>	(1917)
<code>\p{Sentence_Break: Lower}</code>	<code>(Short: \p{SB=LO})</code>	(1917)
<code>\p{Sentence_Break: NU}</code>	<code>\p{Sentence_Break=Numeric}</code>	(412)
<code>\p{Sentence_Break: Numeric}</code>	<code>(Short: \p{SB=NU})</code>	(412)
<code>\p{Sentence_Break: OLetter}</code>	<code>(Short: \p{SB=LE})</code>	(97_369)
<code>\p{Sentence_Break: Other}</code>	<code>(Short: \p{SB=XX})</code>	(1_010_959)
<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>	<code>(3)</code>
<code>\p{Sentence_Break: Sp}</code>	<code>(Short: \p{SB=Sp})</code>	<code>(21)</code>
<code>\p{Sentence_Break: ST}</code>	<code>\p{Sentence_Break=STerm}</code>	<code>(73)</code>
<code>\p{Sentence_Break: STerm}</code>	<code>(Short: \p{SB=ST})</code>	<code>(73)</code>
<code>\p{Sentence_Break: UP}</code>	<code>\p{Sentence_Break=Upper}</code>	<code>(1509)</code>
<code>\p{Sentence_Break: Upper}</code>	<code>(Short: \p{SB=UP})</code>	<code>(1509)</code>
<code>\p{Sentence_Break: XX}</code>	<code>\p{Sentence_Break=Other}</code>	<code>(1_010_959)</code>
<code>\p{Separator}</code>	<code>\p{General_Category=Separator}</code>	<code>(Short: \p{Z})</code>
<code>\p{Shavian}</code>	<code>\p{Script=Shavian}</code>	<code>(Short: \p{Shaw})</code>
<code>\p{Shaw}</code>	<code>\p{Shavian}</code>	<code>(= \p{Script=Shavian})</code>
<code>\p{Sinh}</code>	<code>\p{Sinhala}</code>	<code>(= \p{Script=Sinhala})</code>
	<code>\p{Block=Sinhala}</code>	<code>(80)</code>
<code>\p{Sinhala}</code>	<code>\p{Script=Sinhala}</code>	<code>(Short: \p{Sinh}; NOT \p{Block=Sinhala})</code>
<code>\p{Sk}</code>	<code>\p{Modifier_Symbol}</code>	<code>(= \p{General_Category=Modifier_Symbol})</code>
		<code>(115)</code>
<code>\p{Sm}</code>	<code>\p{Math_Symbol}</code>	<code>(= \p{General_Category=Math_Symbol})</code>
		<code>(948)</code>
X <code>\p{Small_Form_Variants}</code>	<code>\p{Block=Small_Form_Variants}</code>	<code>(32)</code>
<code>\p{So}</code>	<code>\p{Other_Symbol}</code>	<code>(= \p{General_Category=Other_Symbol})</code>
		<code>(4398)</code>
<code>\p{Soft_Dotted}</code>	<code>\p{Soft_Dotted=Y}</code>	<code>(Short: \p{SD})</code>
<code>\p{Soft_Dotted: N*}</code>	<code>(Short: \p{SD=N}, \p{SD})</code>	<code>(1_114_066)</code>
<code>\p{Soft_Dotted: Y*}</code>	<code>(Short: \p{SD=Y}, \p{SD})</code>	<code>(46)</code>
<code>\p{Space}</code>	<code>\p{White_Space=Y}</code>	<code>\s including beyond ASCII plus vertical tab</code>
		<code>(26)</code>
<code>\p{Space: *}</code>	<code>\p{White_Space: *}</code>	
<code>\p{Space_Separator}</code>	<code>\p{General_Category=Space_Separator}</code>	<code>(Short: \p{Zs})</code>
		<code>(18)</code>
<code>\p{SpacePerl}</code>	<code>\p{XPerlSpace}</code>	<code>(25)</code>
<code>\p{Spacing_Mark}</code>	<code>\p{General_Category=Spacing_Mark}</code>	<code>(Short: \p{Mc})</code>
		<code>(287)</code>
X <code>\p{Spacing_Modifier_Letters}</code>	<code>\p{Block=Spacing_Modifier_Letters}</code>	<code>(80)</code>
X <code>\p{Specials}</code>	<code>\p{Block=Specials}</code>	<code>(16)</code>
<code>\p{STerm}</code>	<code>\p{STerm=Y}</code>	<code>(76)</code>
<code>\p{STerm: N*}</code>	<code>(Single: \p{STerm})</code>	<code>(1_114_036)</code>
<code>\p{STerm: Y*}</code>	<code>(Single: \p{STerm})</code>	<code>(76)</code>
<code>\p{Sund}</code>	<code>\p{Sundanese}</code>	<code>(= \p{Script=Sundanese})</code>
	<code>(NOT \p{Block=Sundanese})</code>	<code>(55)</code>
<code>\p{Sundanese}</code>	<code>\p{Script=Sundanese}</code>	<code>(Short: \p{Sund}; NOT \p{Block=Sundanese})</code>
		<code>(55)</code>
X <code>\p{Superscripts_And_Subscripts}</code>	<code>\p{Block=Superscripts_And_Subscripts}</code>	<code>(48)</code>
X <code>\p{Supplemental_Arrows_A}</code>	<code>\p{Block=Supplemental_Arrows_A}</code>	<code>(16)</code>
X <code>\p{Supplemental_Arrows_B}</code>	<code>\p{Block=Supplemental_Arrows_B}</code>	<code>(128)</code>
X <code>\p{Supplemental_Mathematical_Operators}</code>	<code>\p{Block=Supplemental_Mathematical_Operators}</code>	<code>(256)</code>
X <code>\p{Supplemental_Punctuation}</code>	<code>\p{Block=Supplemental_Punctuation}</code>	<code>(128)</code>
X <code>\p{Supplementary_Private_Use_Area_A}</code>	<code>\p{Block=Supplementary_Private_Use_Area_A}</code>	<code>(65_536)</code>

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{Sylo}</code> (= <code>\p{Script=Sylo}</code>) (NOT <code>\p{Block=Sylo}</code>) (44)
	<code>\p{Sylo}</code>	<code>\p{Script=Sylo}</code> (Short: <code>\p{Sylo}</code> ; NOT <code>\p{Block=Sylo}</code>) (44)
	<code>\p{Symbol}</code>	<code>\p{General_Category=Symbol}</code> (Short: <code>\p{S}</code>) (5508)
	<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{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>) (169)
	<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>) (169)
	<code>\p{Terminal_Punctuation: N*}</code>	(Short: <code>\p{Term=N}</code> , <code>\p{Term}</code>) (1_113_943)
	<code>\p{Terminal_Punctuation: Y*}</code>	(Short: <code>\p{Term=Y}</code> , <code>\p{Term}</code>) (169)
	<code>\p{Tfng}</code>	<code>\p{Tifinagh}</code> (= <code>\p{Script=Tifinagh}</code>) (NOT <code>\p{Block=Tifinagh}</code>) (57)
	<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>) (207)
<code>\p{Tibt}</code>	<code>\p{Tibetan}</code> (= <code>\p{Script=Tibetan}</code>) (NOT <code>\p{Block=Tibetan}</code>) (207)
<code>\p{Tifinagh}</code>	<code>\p{Script=Tifinagh}</code> (Short: <code>\p{Tfng}</code> ; NOT <code>\p{Block=Tifinagh}</code>) (57)
<code>\p{Title}</code>	(/i= Cased=Yes) (31)
<code>\p{Titlecase}</code>	<code>\p{Title}</code> (/i= Cased=Yes) (31)
<code>\p{Titlecase_Letter}</code>	<code>\p{General_Category=Titlecase_Letter}</code> (Short: <code>\p{Lt}</code> ; /i= <code>General_Category=Cased_Letter</code>) (31)
X <code>\p{Transport_And_Map_Symbols}</code>	<code>\p{Block=Transport_And_Map_Symbols}</code> (128)
<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_616)
<code>\p{UIdeo: *}</code>	<code>\p{Unified_Ideograph: *}</code>
<code>\p{Unassigned}</code>	<code>\p{General_Category=Unassigned}</code> (Short: <code>\p{Cn}</code>) (865_147)
X <code>\p{Unified_Canadian_Aboriginal_Syllabics}</code>	<code>\p{Block=Unified_Canadian_Aboriginal_Syllabics}</code> (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_616)
<code>\p{Unified_Ideograph: N*}</code>	(Short: <code>\p{UIdeo=N}</code> , <code>\p{UIdeo}</code>) (1_039_496)
<code>\p{Unified_Ideograph: Y*}</code>	(Short: <code>\p{UIdeo=Y}</code> , <code>\p{UIdeo}</code>) (74_616)
<code>\p{Unknown}</code>	<code>\p{Script=Unknown}</code> (Short: <code>\p{Zzzz}</code>) (1_004_663)
<code>\p{Upper}</code>	<code>\p{Uppercase=Y}</code> (/i= Cased=Yes) (1478)
<code>\p{Upper: *}</code>	<code>\p{Uppercase: *}</code>
<code>\p{Uppercase}</code>	<code>\p{Upper}</code> (= <code>\p{Uppercase=Y}</code>) (/i= Cased=Yes) (1478)
<code>\p{Uppercase: N*}</code>	(Short: <code>\p{Upper=N}</code> , <code>\p{Upper}</code> ; /i= Cased=No) (1_112_634)
<code>\p{Uppercase: Y*}</code>	(Short: <code>\p{Upper=Y}</code> , <code>\p{Upper}</code> ; /i= Cased=Yes) (1478)
<code>\p{Uppercase_Letter}</code>	<code>\p{General_Category=Uppercase_Letter}</code> (Short: <code>\p{Lu}</code> ; /i= <code>General_Category=Cased_Letter</code>) (1436)
<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; = <code>\p{Alnum}</code> + <code>\pM</code> + <code>\p{Pc}</code> (102_724)
<code>\p{Word_Break: ALetter}</code>	(Short: <code>\p{WB=LE}</code>) (24_453)
<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>) (1502)
<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> (310)
<code>\p{Word_Break: Katakana}</code>	(Short: <code>\p{WB=KA}</code>) (310)
<code>\p{Word_Break: LE}</code>	<code>\p{Word_Break=ALetter}</code> (24_453)
<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> (411)
<code>\p{Word_Break: Numeric}</code>	(Short: <code>\p{WB=NU}</code>) (411)
<code>\p{Word_Break: Other}</code>	(Short: <code>\p{WB=XX}</code>) (1_087_251)
<code>\p{Word_Break: XX}</code>	<code>\p{Word_Break=Other}</code> (1_087_251)
<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>) (102_656)
<code>\p{XID_Continue: N*}</code>	(Short: <code>\p{XIDC=N}</code> , <code>\P{XIDC}</code>) (1_011_456)
<code>\p{XID_Continue: Y*}</code>	(Short: <code>\p{XIDC=Y}</code> , <code>\P{XIDC}</code>) (102_656)
<code>\p{XID_Start}</code>	<code>\p{XID_Start=Y}</code> (Short: <code>\p{XIDS}</code>) (100_724)
<code>\p{XID_Start: N*}</code>	(Short: <code>\p{XIDS=N}</code> , <code>\P{XIDS}</code>) (1_013_388)
<code>\p{XID_Start: Y*}</code>	(Short: <code>\p{XIDS=Y}</code> , <code>\P{XIDS}</code>) (100_724)
<code>\p{XIDC}</code>	<code>\p{XID_Continue}</code> (= <code>\p{XID_Continue=Y}</code>) (102_656)
<code>\p{XIDC: *}</code>	<code>\p{XID_Continue: *}</code>
<code>\p{XIDS}</code>	<code>\p{XID_Start}</code> (= <code>\p{XID_Start=Y}</code>) (100_724)
<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{XPerlSpace}</code>	<code>\s</code> , including beyond ASCII (Short: <code>\p{SpacePerl}</code>) (25)
<code>\p{XPosixAlnum}</code>	<code>\p{Alnum}</code> (101_959)
<code>\p{XPosixAlpha}</code>	<code>\p{Alpha}</code> (= <code>\p{Alphabetic=Y}</code>) (101_539)
<code>\p{XPosixBlank}</code>	<code>\p{Blank}</code> (19)
<code>\p{XPosixCntrl}</code>	<code>\p{Cntrl}</code> (= <code>\p{General_Category=Control}</code>) (65)
<code>\p{XPosixDigit}</code>	<code>\p{Digit}</code> (= <code>\p{General_Category=</code> <code>Decimal_Number}</code>) (420)
<code>\p{XPosixGraph}</code>	<code>\p{Graph}</code> (246_832)
<code>\p{XPosixLower}</code>	<code>\p{Lower}</code> (= <code>\p{Lowercase=Y}</code>) (/i= Cased= Yes) (1918)
<code>\p{XPosixPrint}</code>	<code>\p{Print}</code> (246_850)
<code>\p{XPosixPunct}</code>	<code>\p{Punct}</code> + ASCII-range <code>\p{Symbol}</code> (607)
<code>\p{XPosixSpace}</code>	<code>\p{Space}</code> (= <code>\p{White_Space=Y}</code>) (26)
<code>\p{XPosixUpper}</code>	<code>\p{Upper}</code> (= <code>\p{Uppercase=Y}</code>) (/i= Cased= Yes) (1478)
<code>\p{XPosixWord}</code>	<code>\p{Word}</code> (102_724)
<code>\p{XPosixXDigit}</code>	<code>\p{XDigit}</code> (= <code>\p{Hex_Digit=Y}</code>) (44)
<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=</code> <code>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=</code> <code>Line_Separator}</code>) (1)
<code>\p{Zp}</code>	<code>\p{Paragraph_Separator}</code> (= <code>\p{General_Category=</code> <code>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>) (6379)
<code>\p{Zzzz}</code>	<code>\p{Unknown}</code> (= <code>\p{Script=Unknown}</code>) (1_004_663)
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>) (1692)
T <code>\p{__CombAbove}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Canonical_Combining_Class=</code> <code>Above}</code>) (320)
T <code>\p{__X_Begin}</code>	(For internal use by Perl, not necessarily stable) (1_113_907)
T <code>\p{__X_Extend}</code>	(For internal use by Perl, not necessarily stable) (1509)
T <code>\p{__X_LV_LVT_V}</code>	(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:

<code>Lowercase_Mapping</code>	<code>lc()</code> and <code>lcfirst()</code>
<code>Titlecase_Mapping</code>	<code>ucfirst()</code>
<code>Uppercase_Mapping</code>	<code>uc()</code>

`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()`, `vianame()`, and `string_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-Unicode 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)

Deprecated by Unicode. These are characters that expand to more than one character in the specified normalization form, but whether they actually take up more bytes or not depends on the encoding being used. For example, a UTF-8 encoded character may expand to a different number of bytes than a UTF-32 encoded character.

Grapheme_Link (Gr_Link)

Deprecated by Unicode: Duplicates `ccc=vr` (`Canonical_Combining_Class=Virama`)

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 downloading the Unicode database from <http://www.unicode.org/Public/> to `$Config{privlib}/unicore/` in the Perl source tree, changing the controlling lists contained in the program `$Config{privlib}/unicore/mktables` and then re-compiling and installing. (`%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. Some, but not all, 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).

Perl reserves the right to change the format and even the existence of any of those files without notice, except the ones that were in existence prior to release 5.13. If those change, a deprecation cycle will be done first. These are:

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

Each of the files in this directory 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:

```
b    binary
c    Perl's internal (Normalize.pm) decomposition mapping
d    single decimal digit
f    floating point number
i    integer
r    rational: an integer or a fraction
s    string
x    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