

!!!!!!! DO NOT EDIT THIS FILE !!!!!!! # This file is machine-generated by lib/unicore/mktables from the Unicode # database, Version 6.2.0. Any changes made here will be lost!

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

NAME

perluniprops - Index of Unicode Version 6.2.0 character properties in Perl

DESCRIPTION

This document provides information about the portion of the Unicode database that deals with character properties, that is the portion that is defined on single code points. (*Other information in the Unicode data base* below briefly mentions other data that Unicode provides.)

Perl can provide access to all non-provisional Unicode character properties, though not all are enabled by default. The omitted ones are the Unihan properties (accessible via the CPAN module *Unicode::Unihan*) and certain deprecated or Unicode-internal properties. (An installation may choose to recompile Perl's tables to change this. See *Unicode character properties that are NOT accepted by Perl*.)

For most purposes, access to Unicode properties from the Perl core is through regular expression matches, as described in the next section. For some special purposes, and to access the properties that are not suitable for regular expression matching, all the Unicode character properties that Perl handles are accessible via the standard *Unicode::UCD* module, as described in the section *Properties accessible through Unicode::UCD*.

Perl also provides some additional extensions and short-cut synonyms for Unicode properties.

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; see *"Other Properties" in perlunicode* for more information on these. There is some detail about Blocks, Scripts, General_Category, and Bidi_Class in *perlunicode*, but to find out about the intricacies of the official Unicode properties, refer to the Unicode standard. A good starting place is <http://www.unicode.org/reports/tr44/>.

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 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:

Stabilized

A property 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

A 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.

A deprecated property may also be stabilized, but this fact is not shown.

Obsolete

Properties marked with an **'O'** in the table are considered (plain) obsolete. Generally this designation is given to properties that Unicode once used for internal purposes (but not any longer).

Some Perl extensions are present for backwards compatibility and are discouraged from being used, but are not obsolete. An **'X'** flags each such entry in the table. Future Unicode versions may force some of these extensions to be removed without warning, replaced by another property with the same name that means something different. Use the equivalent shown instead.

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.

Legend summary:

`*` is a wild-card

(d+) 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, and may not be stable.

NAME	INFO
X <code>\p{Aegean_Numbers}</code>	<code>\p{Block=Aegean_Numbers}</code> (64)
T <code>\p{Age: 1.1}</code>	<code>\p{Age=V1_1}</code> (33_979)
T <code>\p{Age: 2.0}</code>	<code>\p{Age=V2_0}</code> (144_521)
T <code>\p{Age: 2.1}</code>	<code>\p{Age=V2_1}</code> (2)
T <code>\p{Age: 3.0}</code>	<code>\p{Age=V3_0}</code> (10_307)
T <code>\p{Age: 3.1}</code>	<code>\p{Age=V3_1}</code> (44_978)
T <code>\p{Age: 3.2}</code>	<code>\p{Age=V3_2}</code> (1016)
T <code>\p{Age: 4.0}</code>	<code>\p{Age=V4_0}</code> (1226)
T <code>\p{Age: 4.1}</code>	<code>\p{Age=V4_1}</code> (1273)
T <code>\p{Age: 5.0}</code>	<code>\p{Age=V5_0}</code> (1369)
T <code>\p{Age: 5.1}</code>	<code>\p{Age=V5_1}</code> (1624)
T <code>\p{Age: 5.2}</code>	<code>\p{Age=V5_2}</code> (6648)
T <code>\p{Age: 6.0}</code>	<code>\p{Age=V6_0}</code> (2088)
T <code>\p{Age: 6.1}</code>	<code>\p{Age=V6_1}</code> (732)
T <code>\p{Age: 6.2}</code>	<code>\p{Age=V6_2}</code> (1)
<code>\p{Age: NA}</code>	<code>\p{Age=Unassigned}</code> (864_348)
<code>\p{Age: Unassigned}</code>	Code point's usage has not been assigned in any Unicode release thus far. (Short: <code>\p{Age=NA}</code>) (864_348)
<code>\p{Age: V1_1}</code>	Code point's usage introduced in version 1.1 (33_979)
<code>\p{Age: V2_0}</code>	Code point's usage was introduced in version 2.0; See also Property 'Present_In' (144_521)
<code>\p{Age: V2_1}</code>	Code point's usage was introduced in version 2.1; See also Property 'Present_In' (2)
<code>\p{Age: V3_0}</code>	Code point's usage was introduced in version 3.0; See also Property 'Present_In' (10_307)
<code>\p{Age: V3_1}</code>	Code point's usage was introduced in version 3.1; See also Property 'Present_In' (44_978)
<code>\p{Age: V3_2}</code>	Code point's usage was introduced in version 3.2; See also Property 'Present_In' (1016)
<code>\p{Age: V4_0}</code>	Code point's usage was introduced in version 4.0; See also Property 'Present_In' (1226)
<code>\p{Age: V4_1}</code>	Code point's usage was introduced in version 4.1; See also Property 'Present_In' (1273)
<code>\p{Age: V5_0}</code>	Code point's usage was introduced in version 5.0; See also Property 'Present_In' (1369)
<code>\p{Age: V5_1}</code>	Code point's usage was introduced in version 5.1; See also Property 'Present_In' (1624)
<code>\p{Age: V5_2}</code>	Code point's usage was introduced in version 5.2; See also Property 'Present_In' (6648)

<code>\p{Age: V6_0}</code>	Code point's usage was introduced in version 6.0; See also Property 'Present_In' (2088)
<code>\p{Age: V6_1}</code>	Code point's usage was introduced in version 6.1; See also Property 'Present_In' (732)
<code>\p{Age: V6_2}</code>	Code point's usage was introduced in version 6.2; See also Property 'Present_In' (1)
<code>\p{AHex}</code>	<code>\p{PosixXDigit}</code> (= <code>\p{ASCII_Hex_Digit=Y}</code>) (22)
<code>\p{AHex: *}</code>	<code>\p{ASCII_Hex_Digit: *}</code>
X <code>\p{Alchemical}</code>	<code>\p{Alchemical_Symbols}</code> (= <code>\p{Block=Alchemical_Symbols}</code>) (128)
X <code>\p{Alchemical_Symbols}</code>	<code>\p{Block=Alchemical_Symbols}</code> (Short: <code>\p{InAlchemical}</code>) (128)
<code>\p{All}</code>	<code>\p{Any}</code> (1_114_112)
<code>\p{Alnum}</code>	Alphabetic and (decimal) Numeric (102_619)
<code>\p{Alpha}</code>	<code>\p{Alphabetic=Y}</code> (102_159)
<code>\p{Alpha: *}</code>	<code>\p{Alphabetic: *}</code>
<code>\p{Alphabetic}</code>	<code>\p{Alpha}</code> (= <code>\p{Alphabetic=Y}</code>) (102_159)
<code>\p{Alphabetic: N*}</code>	(Short: <code>\p{Alpha=N}</code> , <code>\p{Alpha}</code>) (1_011_953)
<code>\p{Alphabetic: Y*}</code>	(Short: <code>\p{Alpha=Y}</code> , <code>\p{Alpha}</code>) (102_159)
X <code>\p{Alphabetic_PF}</code>	<code>\p{Alphabetic_Presentation_Forms}</code> (= <code>\p{Block=Alphabetic_Presentation_Forms}</code>) (80)
X <code>\p{Alphabetic_Presentation_Forms}</code>	<code>\p{Block=Alphabetic_Presentation_Forms}</code> (Short: <code>\p{InAlphabeticPF}</code>) (80)
X <code>\p{Ancient_Greek_Music}</code>	<code>\p{Ancient_Greek_Musical_Notation}</code> (= <code>\p{Block=Ancient_Greek_Musical_Notation}</code>) (80)
X <code>\p{Ancient_Greek_Musical_Notation}</code>	<code>\p{Block=Ancient_Greek_Musical_Notation}</code> (Short: <code>\p{InAncientGreekMusic}</code>) (80)
X <code>\p{Ancient_Greek_Numbers}</code>	<code>\p{Block=Ancient_Greek_Numbers}</code> (80)
X <code>\p{Ancient_Symbols}</code>	<code>\p{Block=Ancient_Symbols}</code> (64)
<code>\p{Any}</code>	<code>[\x{0000}-\x{10FFFF}]</code> (1_114_112)
<code>\p{Arab}</code>	<code>\p{Arabic}</code> (= <code>\p{Script=Arabic}</code>) (NOT <code>\p{Block=Arabic}</code>) (1235)
<code>\p{Arabic}</code>	<code>\p{Script=Arabic}</code> (Short: <code>\p{Arab}</code> ; NOT <code>\p{Block=Arabic}</code>) (1235)
X <code>\p{Arabic_Ext_A}</code>	<code>\p{Arabic_Extended_A}</code> (= <code>\p{Block=Arabic_Extended_A}</code>) (96)
X <code>\p{Arabic_Extended_A}</code>	<code>\p{Block=Arabic_Extended_A}</code> (Short: <code>\p{InArabicExtA}</code>) (96)
X <code>\p{Arabic_Math}</code>	<code>\p{Arabic_Mathematical_Alphabetic_Symbols}</code> (= <code>\p{Block=Arabic_Mathematical_Alphabetic_Symbols}</code>) (256)
X <code>\p{Arabic_Mathematical_Alphabetic_Symbols}</code>	<code>\p{Block=Arabic_Mathematical_Alphabetic_Symbols}</code> (Short: <code>\p{InArabicMath}</code>) (256)
X <code>\p{Arabic_PF_A}</code>	<code>\p{Arabic_Presentation_Forms_A}</code> (= <code>\p{Block=Arabic_Presentation_Forms_A}</code>) (688)

X	<code>\p{Arabic_PF_B}</code>	<code>\p{Arabic_Presentation_Forms_B}</code> (= <code>\p{Block=Arabic_Presentation_Forms_B}</code>) (144)
X	<code>\p{Arabic_Presentation_Forms_A}</code>	<code>\p{Block=Arabic_Presentation_Forms_A}</code> (Short: <code>\p{InArabicPFA}</code>) (688)
X	<code>\p{Arabic_Presentation_Forms_B}</code>	<code>\p{Block=Arabic_Presentation_Forms_B}</code> (Short: <code>\p{InArabicPFB}</code>) (144)
X	<code>\p{Arabic_Sup}</code>	<code>\p{Arabic_Supplement}</code> (= <code>\p{Block=Arabic_Supplement}</code>) (48)
X	<code>\p{Arabic_Supplement}</code>	<code>\p{Block=Arabic_Supplement}</code> (Short: <code>\p{InArabicSup}</code>) (48)
	<code>\p{Armenian}</code>	<code>\p{Script=Armenian}</code> (Short: <code>\p{Armn}</code>); NOT <code>\p{Block=Armenian}</code>) (91)
	<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>) (91)
X	<code>\p{Arrows}</code>	<code>\p{Block=Arrows}</code> (112)
	<code>\p{ASCII}</code>	<code>\p{Block=Basic_Latin}</code> [[:ASCII:]] (128)
	<code>\p{ASCII_Hex_Digit}</code>	<code>\p{PosixXDigit}</code> (= <code>\p{ASCII_Hex_Digit=Y}</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 (249_698)
	<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_Sup}</code>	<code>\p{Bamum_Supplement}</code> (= <code>\p{Block=Bamum_Supplement}</code>) (576)
X	<code>\p{Bamum_Supplement}</code>	<code>\p{Block=Bamum_Supplement}</code> (Short: <code>\p{InBamumSup}</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> (1438)

```

\p{Bidi_Class: AN}          \p{Bidi_Class=Arabic_Number} (49)
\p{Bidi_Class: Arabic_Letter} (Short: \p{Bc=AL}) (1438)
\p{Bidi_Class: Arabic_Number} (Short: \p{Bc=AN}) (49)
\p{Bidi_Class: B}          \p{Bidi_Class=Paragraph_Separator} (7)
\p{Bidi_Class: BN}        \p{Bidi_Class=Boundary_Neutral} (4015)
\p{Bidi_Class: Boundary_Neutral} (Short: \p{Bc=BN}) (4015)
\p{Bidi_Class: Common_Separator} (Short: \p{Bc=CS}) (15)
\p{Bidi_Class: CS}        \p{Bidi_Class=Common_Separator} (15)
\p{Bidi_Class: EN}        \p{Bidi_Class=European_Number} (131)
\p{Bidi_Class: ES}        \p{Bidi_Class=European_Separator} (12)
\p{Bidi_Class: ET}        \p{Bidi_Class=European_Terminator} (66)
\p{Bidi_Class: European_Number} (Short: \p{Bc=EN}) (131)
\p{Bidi_Class: European_Separator} (Short: \p{Bc=ES}) (12)
\p{Bidi_Class: European_Terminator} (Short: \p{Bc=ET}) (66)
\p{Bidi_Class: L}         \p{Bidi_Class=Left_To_Right} (1_098_530)
\p{Bidi_Class: Left_To_Right} (Short: \p{Bc=L}) (1_098_530)
\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}) (1290)
\p{Bidi_Class: NSM}       \p{Bidi_Class=Nonspacing_Mark} (1290)
\p{Bidi_Class: ON}        \p{Bidi_Class=Other_Neutral} (4447)
\p{Bidi_Class: Other_Neutral} (Short: \p{Bc=ON}) (4447)
\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} (4086)
\p{Bidi_Class: Right_To_Left} (Short: \p{Bc=R}) (4086)
\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})
                           (545)
\p{Bidi_M: *}             \p{Bidi_Mirrored: *}
\p{Bidi_Mirrored}        \p{Bidi_Mirrored=Y} (Short: \p{BidiM})
                           (545)
\p{Bidi_Mirrored: N*}    (Short: \p{BidiM=N}, \p{BidiM}) (1_113_567)
\p{Bidi_Mirrored: Y*}    (Short: \p{BidiM=Y}, \p{BidiM}) (545)
\p{Blank}                \h, Horizontal white space (19)
\p{Blk: *}                \p{Block: *}
\p{Block: Aegean_Numbers} (Single: \p{InAegeanNumbers}) (64)
\p{Block: Alchemical}     \p{Block=Alchemical_Symbols} (128)
\p{Block: Alchemical_Symbols} (Short: \p{Blk=Alchemical},
                               \p{InAlchemical}) (128)
\p{Block: Alphabetic_PF} \p{Block=Alphabetic_Presentation_Forms}
                           (80)
\p{Block: Alphabetic_Presentation_Forms} (Short: \p{Blk=

```

AlphabeticPF}, \p{InAlphabeticPF}) (80)

\p{Block: Ancient_Greek_Music} \p{Block= Ancient_Greek_Musical_Notation} (80)

\p{Block: Ancient_Greek_Musical_Notation} (Short: \p{Blk= AncientGreekMusic}, \p{InAncientGreekMusic}) (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_Ext_A} \p{Block=Arabic_Extended_A} (96)

\p{Block: Arabic_Extended_A} (Short: \p{Blk=ArabicExtA}, \p{InArabicExtA}) (96)

\p{Block: Arabic_Math} \p{Block= Arabic_Mathematical_Alphabetic_Symbols} (256)

\p{Block: Arabic_Mathematical_Alphabetic_Symbols} (Short: \p{Blk= ArabicMath}, \p{InArabicMath}) (256)

\p{Block: Arabic_PF_A} \p{Block=Arabic_Presentation_Forms_A} (688)

\p{Block: Arabic_PF_B} \p{Block=Arabic_Presentation_Forms_B} (144)

\p{Block: Arabic_Presentation_Forms_A} (Short: \p{Blk=ArabicPFA}, \p{InArabicPFA}) (688)

\p{Block: Arabic_Presentation_Forms_B} (Short: \p{Blk=ArabicPFB}, \p{InArabicPFB}) (144)

\p{Block: Arabic_Sup} \p{Block=Arabic_Supplement} (48)

\p{Block: Arabic_Supplement} (Short: \p{Blk=ArabicSup}, \p{InArabicSup}) (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_Sup} \p{Block=Bamum_Supplement} (576)

\p{Block: Bamum_Supplement} (Short: \p{Blk=BamumSup}, \p{InBamumSup}) (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_Ext} \p{Block=Bopomofo_Extended} (32)

\p{Block: Bopomofo_Extended} (Short: \p{Blk=BopomofoExt}, \p{InBopomofoExt}) (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} \p{Block=Braille_Patterns} (256)

```

\p{Block: Braille_Patterns} (Short: \p{Blk=Braille},
                             \p{InBraille}) (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_Music} \p{Block=Byzantine_Musical_Symbols}
                             (256)
\p{Block: Byzantine_Musical_Symbols} (Short: \p{Blk=
                                         ByzantineMusic}, \p{InByzantineMusic})
                                         (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: Chakma}         (Single: \p{InChakma}; NOT \p{Chakma} NOR
                             \p{Is_Chakma}) (80)
\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}            \p{Block=CJK_Unified_Ideographs} (20_992)
\p{Block: CJK_Compat}     \p{Block=CJK_Compatibility} (256)
\p{Block: CJK_Compat_Forms} \p{Block=CJK_Compatibility_Forms} (32)
\p{Block: CJK_Compat_Ideographs} \p{Block=
                                         CJK_Compatibility_Ideographs} (512)
\p{Block: CJK_Compat_Ideographs_Sup} \p{Block=
                                         CJK_Compatibility_Ideographs_Supplement}
                                         (544)
\p{Block: CJK_Compatibility} (Short: \p{Blk=CJKCompat},
                                \p{InCJKCompat}) (256)
\p{Block: CJK_Compatibility_Forms} (Short: \p{Blk=CJKCompatForms},
                                         \p{InCJKCompatForms}) (32)
\p{Block: CJK_Compatibility_Ideographs} (Short: \p{Blk=
                                         CJKCompatIdeographs},
                                         \p{InCJKCompatIdeographs}) (512)
\p{Block: CJK_Compatibility_Ideographs_Supplement} (Short: \p{Blk=
                                         CJKCompatIdeographsSup},
                                         \p{InCJKCompatIdeographsSup}) (544)
\p{Block: CJK_Ext_A}      \p{Block=
                                         CJK_Unified_Ideographs_Extension_A}
                                         (6592)
\p{Block: CJK_Ext_B}      \p{Block=
                                         CJK_Unified_Ideographs_Extension_B}
                                         (42_720)
\p{Block: CJK_Ext_C}      \p{Block=
                                         CJK_Unified_Ideographs_Extension_C}
                                         (4160)
\p{Block: CJK_Ext_D}      \p{Block=
                                         CJK_Unified_Ideographs_Extension_D} (224)
\p{Block: CJK_Radicals_Sup} \p{Block=CJK_Radicals_Supplement} (128)
\p{Block: CJK_Radicals_Supplement} (Short: \p{Blk=CJKRadicalsSup},
                                       \p{InCJKRadicalsSup}) (128)
\p{Block: CJK_Strokes}   (Single: \p{InCJKStrokes}) (48)
\p{Block: CJK_Symbols}   \p{Block=CJK_Symbols_And_Punctuation} (64)
    
```

`\p{Block: CJK_Symbols_And_Punctuation}` (Short: `\p{Blk=CJKSymbols}`,
`\p{InCJKSymbols}`) (64)
`\p{Block: CJK_Unified_Ideographs}` (Short: `\p{Blk=CJK}`, `\p{InCJK}`)
(20_992)
`\p{Block: CJK_Unified_Ideographs_Extension_A}` (Short: `\p{Blk=`
`CJKExtA}`, `\p{InCJKExtA}`) (6592)
`\p{Block: CJK_Unified_Ideographs_Extension_B}` (Short: `\p{Blk=`
`CJKExtB}`, `\p{InCJKExtB}`) (42_720)
`\p{Block: CJK_Unified_Ideographs_Extension_C}` (Short: `\p{Blk=`
`CJKExtC}`, `\p{InCJKExtC}`) (4160)
`\p{Block: CJK_Unified_Ideographs_Extension_D}` (Short: `\p{Blk=`
`CJKExtD}`, `\p{InCJKExtD}`) (224)
`\p{Block: Combining_Diacritical_Marks}` (Short: `\p{Blk=`
`Diacriticals}`, `\p{InDiacriticals}`) (112)
`\p{Block: Combining_Diacritical_Marks_For_Symbols}` (Short: `\p{Blk=`
`DiacriticalsForSymbols}`,
`\p{InDiacriticalsForSymbols}`) (48)
`\p{Block: Combining_Diacritical_Marks_Supplement}` (Short: `\p{Blk=`
`DiacriticalsSup}`, `\p{InDiacriticalsSup}`)
(64)
`\p{Block: Combining_Half_Marks}` (Short: `\p{Blk=HalfMarks}`,
`\p{InHalfMarks}`) (16)
`\p{Block: Combining_Marks_For_Symbols}` `\p{Block=`
`Combining_Diacritical_Marks_For_Symbols}`
(48)
`\p{Block: Common_Indic_Number_Forms}` (Short: `\p{Blk=`
`IndicNumberForms}`,
`\p{InIndicNumberForms}`) (16)
`\p{Block: Compat_Jamo}` `\p{Block=Hangul_Compatibility_Jamo}` (96)
`\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}` `\p{Block=Counting_Rod_Numerals}` (32)
`\p{Block: Counting_Rod_Numerals}` (Short: `\p{Blk=CountingRod}`,
`\p{InCountingRod}`) (32)
`\p{Block: Cuneiform}` (Single: `\p{InCuneiform}`; NOT
`\p{Cuneiform}` NOR `\p{Is_Cuneiform}`)
(1024)
`\p{Block: Cuneiform_Numbers}` `\p{Block=`
`Cuneiform_Numbers_And_Punctuation}` (128)
`\p{Block: Cuneiform_Numbers_And_Punctuation}` (Short: `\p{Blk=`
`CuneiformNumbers}`,
`\p{InCuneiformNumbers}`) (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_Ext_A}` `\p{Block=Cyrillic_Extended_A}` (32)
`\p{Block: Cyrillic_Ext_B}` `\p{Block=Cyrillic_Extended_B}` (96)
`\p{Block: Cyrillic_Extended_A}` (Short: `\p{Blk=CyrillicExtA}`,
`\p{InCyrillicExtA}`) (32)
`\p{Block: Cyrillic_Extended_B}` (Short: `\p{Blk=CyrillicExtB}`,
`\p{InCyrillicExtB}`) (96)
`\p{Block: Cyrillic_Sup}` `\p{Block=Cyrillic_Supplement}` (48)
`\p{Block: Cyrillic_Supplement}` (Short: `\p{Blk=CyrillicSup}`,
`\p{InCyrillicSup}`) (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_Ext} \p{Block=Devanagari_Extended} (32)
\p{Block: Devanagari_Extended} (Short: \p{Blk=DevanagariExt},
    \p{InDevanagariExt}) (32)
\p{Block: Diacriticals} \p{Block=Combining_Diacritical_Marks} (112)
\p{Block: Diacriticals_For_Symbols} \p{Block=
    Combining_Diacritical_Marks_For_Symbols}
    (48)
\p{Block: Diacriticals_Sup} \p{Block=
    Combining_Diacritical_Marks_Supplement}
    (64)
\p{Block: Dingbats}      (Single: \p{InDingbats}) (192)
\p{Block: Domino}        \p{Block=Domino_Tiles} (112)
\p{Block: Domino_Tiles} (Short: \p{Blk=Domino}, \p{InDomino}) (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_Alphanum} \p{Block=Enclosed_Alphanumerics} (160)
\p{Block: Enclosed_Alphanum_Sup} \p{Block=
    Enclosed_Alphanumeric_Supplement} (256)
\p{Block: Enclosed_Alphanumeric_Supplement} (Short: \p{Blk=
    EnclosedAlphanumSup},
    \p{InEnclosedAlphanumSup}) (256)
\p{Block: Enclosed_Alphanumerics} (Short: \p{Blk=
    EnclosedAlphanum},
    \p{InEnclosedAlphanum}) (160)
\p{Block: Enclosed_CJK} \p{Block=Enclosed_CJK_Letters_And_Months}
    (256)
\p{Block: Enclosed_CJK_Letters_And_Months} (Short: \p{Blk=
    EnclosedCJK}, \p{InEnclosedCJK}) (256)
\p{Block: Enclosed_Ideographic_Sup} \p{Block=
    Enclosed_Ideographic_Supplement} (256)
\p{Block: Enclosed_Ideographic_Supplement} (Short: \p{Blk=
    EnclosedIdeographicSup},
    \p{InEnclosedIdeographicSup}) (256)
\p{Block: Ethiopic}      (Single: \p{InEthiopic}; NOT \p{Ethiopic}
    NOR \p{Is_Ethiopic}) (384)
\p{Block: Ethiopic_Ext} \p{Block=Ethiopic_Extended} (96)
\p{Block: Ethiopic_Ext_A} \p{Block=Ethiopic_Extended_A} (48)
\p{Block: Ethiopic_Extended} (Short: \p{Blk=EthiopicExt},
    \p{InEthiopicExt}) (96)
\p{Block: Ethiopic_Extended_A} (Short: \p{Blk=EthiopicExtA},
    \p{InEthiopicExtA}) (48)
\p{Block: Ethiopic_Sup} \p{Block=Ethiopic_Supplement} (32)
\p{Block: Ethiopic_Supplement} (Short: \p{Blk=EthiopicSup},
    \p{InEthiopicSup}) (32)
\p{Block: General_Punctuation} (Short: \p{Blk=Punctuation},
    \p{InPunctuation}; NOT \p{Punct} NOR
    \p{Is_Punctuation}) (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_Sup}` `\p{Block=Georgian_Supplement}` (48)
`\p{Block: Georgian_Supplement}` (Short: `\p{Blk=GeorgianSup}`,
`\p{InGeorgianSup}`) (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_Ext}` `\p{Block=Greek_Extended}` (256)
`\p{Block: Greek_Extended}` (Short: `\p{Blk=GreekExt}`,
`\p{InGreekExt}`) (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: Half_And_Full_Forms}` `\p{Block=`
`Halfwidth_And_Fullwidth_Forms}` (240)
`\p{Block: Half_Marks}` `\p{Block=Combining_Half_Marks}` (16)
`\p{Block: Halfwidth_And_Fullwidth_Forms}` (Short: `\p{Blk=`
`HalfAndFullForms}`,
`\p{InHalfAndFullForms}`) (240)
`\p{Block: Hangul}` `\p{Block=Hangul_Syllables}` (NOT `\p{Hangul}`
NOR `\p{Is_Hangul}`) (11_184)
`\p{Block: Hangul_Compatibility_Jamo}` (Short: `\p{Blk=CompatJamo}`,
`\p{InCompatJamo}`) (96)
`\p{Block: Hangul_Jamo}` (Short: `\p{Blk=Jamo}`, `\p{InJamo}`) (256)
`\p{Block: Hangul_Jamo_Extended_A}` (Short: `\p{Blk=JamoExtA}`,
`\p{InJamoExtA}`) (32)
`\p{Block: Hangul_Jamo_Extended_B}` (Short: `\p{Blk=JamoExtB}`,
`\p{InJamoExtB}`) (80)
`\p{Block: Hangul_Syllables}` (Short: `\p{Blk=Hangul}`, `\p{InHangul}`;
NOT `\p{Hangul}` NOR `\p{Is_Hangul}`)
(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}` (Short: `\p{Blk=`
`HighPUSurrogates}`,
`\p{InHighPUSurrogates}`) (128)
`\p{Block: High_PU_Surrogates}` `\p{Block=`
`High_Private_Use_Surrogates}` (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: IDC}` `\p{Block=`
`Ideographic_Description_Characters}` (NOT
`\p{ID_Continue}` NOR `\p{Is_IDC}`) (16)
`\p{Block: Ideographic_Description_Characters}` (Short: `\p{Blk=IDC}`,

```

        \p{InIDC}; NOT \p{ID_Continue} NOR
        \p{Is_IDC}) (16)
\p{Block: Imperial_Aramaic} (Single: \p{InImperialAramaic}; NOT
        \p{Imperial_Aramaic} NOR
        \p{Is_Imperial_Aramaic}) (32)
\p{Block: Indic_Number_Forms} \p{Block=Common_Indic_Number_Forms}
        (16)
\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_Ext} \p{Block=IPA_Extensions} (96)
\p{Block: IPA_Extensions} (Short: \p{Blk=IPAExt}, \p{InIPAExt})
        (96)
\p{Block: Jamo} \p{Block=Hangul_Jamo} (256)
\p{Block: Jamo_Ext_A} \p{Block=Hangul_Jamo_Extended_A} (32)
\p{Block: Jamo_Ext_B} \p{Block=Hangul_Jamo_Extended_B} (80)
\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_Sup} \p{Block=Kana_Supplement} (256)
\p{Block: Kana_Supplement} (Short: \p{Blk=KanaSup}, \p{InKanaSup})
        (256)
\p{Block: Kanbun} (Single: \p{InKanbun}) (16)
\p{Block: Kangxi} \p{Block=Kangxi_Radicals} (224)
\p{Block: Kangxi_Radicals} (Short: \p{Blk=Kangxi}, \p{InKangxi})
        (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_Ext} \p{Block=Katakana_Phonetic_Extensions} (16)
\p{Block: Katakana_Phonetic_Extensions} (Short: \p{Blk=
        KatakanaExt}, \p{InKatakanaExt}) (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_Sup} \p{Block=Latin_1_Supplement} (128)
\p{Block: Latin_1_Supplement} (Short: \p{Blk=Latin1},
        \p{InLatin1}) (128)
\p{Block: Latin_Ext_A} \p{Block=Latin_Extended_A} (128)
\p{Block: Latin_Ext_Additional} \p{Block=
        Latin_Extended_Additional} (256)
\p{Block: Latin_Ext_B} \p{Block=Latin_Extended_B} (208)
    
```

```

\p{Block: Latin_Ext_C} \p{Block=Latin_Extended_C} (32)
\p{Block: Latin_Ext_D} \p{Block=Latin_Extended_D} (224)
\p{Block: Latin_Extended_A} (Short: \p{Blk=LatinExtA},
    \p{InLatinExtA}) (128)
\p{Block: Latin_Extended_Additional} (Short: \p{Blk=
    LatinExtAdditional},
    \p{InLatinExtAdditional}) (256)
\p{Block: Latin_Extended_B} (Short: \p{Blk=LatinExtB},
    \p{InLatinExtB}) (208)
\p{Block: Latin_Extended_C} (Short: \p{Blk=LatinExtC},
    \p{InLatinExtC}) (32)
\p{Block: Latin_Extended_D} (Short: \p{Blk=LatinExtD},
    \p{InLatinExtD}) (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} \p{Block=Mahjong_Tiles} (48)
\p{Block: Mahjong_Tiles} (Short: \p{Blk=Mahjong}, \p{InMahjong})
    (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: Math_Alphanum} \p{Block=
    Mathematical_Alphanumeric_Symbols} (1024)
\p{Block: Math_Operators} \p{Block=Mathematical_Operators} (256)
\p{Block: Mathematical_Alphanumeric_Symbols} (Short: \p{Blk=
    MathAlphanum}, \p{InMathAlphanum}) (1024)
\p{Block: Mathematical_Operators} (Short: \p{Blk=MathOperators},
    \p{InMathOperators}) (256)
\p{Block: Meetei_Mayek} (Single: \p{InMeeteiMayek}; NOT
    \p{Meetei_Mayek} NOR
    \p{Is_Meetei_Mayek}) (64)
\p{Block: Meetei_Mayek_Ext} \p{Block=Meetei_Mayek_Extensions} (32)
\p{Block: Meetei_Mayek_Extensions} (Short: \p{Blk=MeeteiMayekExt},
    \p{InMeeteiMayekExt}) (32)
\p{Block: Meroitic_Cursive} (Single: \p{InMeroiticCursive}; NOT
    \p{Meroitic_Cursive} NOR
    \p{Is_Meroitic_Cursive}) (96)
\p{Block: Meroitic_Hieroglyphs} (Single:
    \p{InMeroiticHieroglyphs}) (32)
\p{Block: Miao} (Single: \p{InMiao}; NOT \p{Miao} NOR
    \p{Is_Miao}) (160)
    
```

```

\p{Block: Misc_Arrows} \p{Block=Miscellaneous_Symbols_And_Arrows}
(256)
\p{Block: Misc_Math_Symbols_A} \p{Block=
Miscellaneous_Mathematical_Symbols_A}
(48)
\p{Block: Misc_Math_Symbols_B} \p{Block=
Miscellaneous_Mathematical_Symbols_B}
(128)
\p{Block: Misc_Pictographs} \p{Block=
Miscellaneous_Symbols_And_Pictographs}
(768)
\p{Block: Misc_Symbols} \p{Block=Miscellaneous_Symbols} (256)
\p{Block: Misc_Technical} \p{Block=Miscellaneous_Technical} (256)
\p{Block: Miscellaneous_Mathematical_Symbols_A} (Short: \p{Blk=
MiscMathSymbolsA},
\p{InMiscMathSymbolsA}) (48)
\p{Block: Miscellaneous_Mathematical_Symbols_B} (Short: \p{Blk=
MiscMathSymbolsB},
\p{InMiscMathSymbolsB}) (128)
\p{Block: Miscellaneous_Symbols} (Short: \p{Blk=MiscSymbols},
\p{InMiscSymbols}) (256)
\p{Block: Miscellaneous_Symbols_And_Arrows} (Short: \p{Blk=
MiscArrows}, \p{InMiscArrows}) (256)
\p{Block: Miscellaneous_Symbols_And_Pictographs} (Short: \p{Blk=
MiscPictographs}, \p{InMiscPictographs})
(768)
\p{Block: Miscellaneous_Technical} (Short: \p{Blk=MiscTechnical},
\p{InMiscTechnical}) (256)
\p{Block: Modifier_Letters} \p{Block=Spacing_Modifier_Letters} (80)
\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: Music} \p{Block=Musical_Symbols} (256)
\p{Block: Musical_Symbols} (Short: \p{Blk=Music}, \p{InMusic})
(256)
\p{Block: Myanmar} (Single: \p{InMyanmar}; NOT \p{Myanmar}
NOR \p{Is_Myanmar}) (160)
\p{Block: Myanmar_Ext_A} \p{Block=Myanmar_Extended_A} (32)
\p{Block: Myanmar_Extended_A} (Short: \p{Blk=MyanmarExtA},
\p{InMyanmarExtA}) (32)
\p{Block: NB} \p{Block=No_Block} (860_672)
\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} (Short: \p{Blk=NB}, \p{InNB}) (860_672)
\p{Block: Number_Forms} (Single: \p{InNumberForms}) (64)
\p{Block: OCR} \p{Block=Optical_Character_Recognition}
(32)
\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} (Short: \p{Blk=OCR},
\p{InOCR}) (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} \p{Block=Phaistos_Disc} (48)
\p{Block: Phaistos_Disc} (Short: \p{Blk=Phaistos}, \p{InPhaistos})
(48)
\p{Block: Phoenician} (Single: \p{InPhoenician}; NOT
\p{Phoenician} NOR \p{Is_Phoenician})
(32)
\p{Block: Phonetic_Ext} \p{Block=Phonetic_Extensions} (128)
\p{Block: Phonetic_Ext_Sup} \p{Block=
Phonetic_Extensions_Supplement} (64)
\p{Block: Phonetic_Extensions} (Short: \p{Blk=PhoneticExt},
\p{InPhoneticExt}) (128)
\p{Block: Phonetic_Extensions_Supplement} (Short: \p{Blk=
PhoneticExtSup}, \p{InPhoneticExtSup})
(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=PUA}, \p{InPUA}; NOT
\p{Private_Use} NOR \p{Is_Private_Use})
(6400)
\p{Block: PUA} \p{Block=Private_Use_Area} (NOT
\p{Private_Use} NOR \p{Is_Private_Use})
(6400)
\p{Block: Punctuation} \p{Block=General_Punctuation} (NOT
\p{Punct} NOR \p{Is_Punctuation}) (112)
\p{Block: Rejang} (Single: \p{InRejang}; NOT \p{Rejang} NOR
\p{Is_Rejang}) (48)
\p{Block: Rumi} \p{Block=Rumi_Numeral_Symbols} (32)
\p{Block: Rumi_Numeral_Symbols} (Short: \p{Blk=Rumi}, \p{InRumi})
(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: Sharada} (Single: \p{InSharada}; NOT \p{Sharada}
NOR \p{Is_Sharada}) (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}` (Short: `\p{Blk=SmallForms}`,
`\p{InSmallForms}`) (32)
`\p{Block: Small_Forms}` `\p{Block=Small_Form_Variants}` (32)
`\p{Block: Sora_Sompeng}` (Single: `\p{InSoraSompeng}`; NOT
`\p{Sora_Sompeng}` NOR
`\p{Is_Sora_Sompeng}`) (48)
`\p{Block: Spacing_Modifier_Letters}` (Short: `\p{Blk=`
`ModifierLetters}`, `\p{InModifierLetters}`)
 (80)
`\p{Block: Specials}` (Single: `\p{InSpecials}`) (16)
`\p{Block: Sundanese}` (Single: `\p{InSundanese}`; NOT
`\p{Sundanese}` NOR `\p{Is_Sundanese}`) (64)
`\p{Block: Sundanese_Sup}` `\p{Block=Sundanese_Supplement}` (16)
`\p{Block: Sundanese_Supplement}` (Short: `\p{Blk=SundaneseSup}`,
`\p{InSundaneseSup}`) (16)
`\p{Block: Sup_Arrows_A}` `\p{Block=Supplemental_Arrows_A}` (16)
`\p{Block: Sup_Arrows_B}` `\p{Block=Supplemental_Arrows_B}` (128)
`\p{Block: Sup_Math_Operators}` `\p{Block=`
`Supplemental_Mathematical_Operators}`
 (256)
`\p{Block: Sup_PUA_A}` `\p{Block=Supplementary_Private_Use_Area_A}`
 (65_536)
`\p{Block: Sup_PUA_B}` `\p{Block=Supplementary_Private_Use_Area_B}`
 (65_536)
`\p{Block: Sup_Punctuation}` `\p{Block=Supplemental_Punctuation}` (128)
`\p{Block: Super_And_Sub}` `\p{Block=Superscripts_And_Subscripts}` (48)
`\p{Block: Superscripts_And_Subscripts}` (Short: `\p{Blk=`
`SuperAndSub}`, `\p{InSuperAndSub}`) (48)
`\p{Block: Supplemental_Arrows_A}` (Short: `\p{Blk=SupArrowsA}`,
`\p{InSupArrowsA}`) (16)
`\p{Block: Supplemental_Arrows_B}` (Short: `\p{Blk=SupArrowsB}`,
`\p{InSupArrowsB}`) (128)
`\p{Block: Supplemental_Mathematical_Operators}` (Short: `\p{Blk=`
`SupMathOperators}`,
`\p{InSupMathOperators}`) (256)
`\p{Block: Supplemental_Punctuation}` (Short: `\p{Blk=`
`SupPunctuation}`, `\p{InSupPunctuation}`)
 (128)
`\p{Block: Supplementary_Private_Use_Area_A}` (Short: `\p{Blk=`
`SupPUAA}`, `\p{InSupPUAA}`) (65_536)
`\p{Block: Supplementary_Private_Use_Area_B}` (Short: `\p{Blk=`
`SupPUAB}`, `\p{InSupPUAB}`) (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{Block: Tai_Tham}      (\p{Is_Tai_Le}) (48)
\p{Block: Tai_Viet}     (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} \p{Block=Tai_Xuan_Jing_Symbols} (96)
\p{Block: Tai_Xuan_Jing_Symbols} (Short: \p{Blk=TaiXuanJing},
\p{InTaiXuanJing}) (96)
\p{Block: Takri}        (Single: \p{InTakri}; NOT \p{Takri} NOR
                        \p{Is_Takri}) (80)
\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} \p{Block=Transport_And_Map_Symbols}
                        (128)
\p{Block: Transport_And_Map_Symbols} (Short: \p{Blk=
                        TransportAndMap}, \p{InTransportAndMap})
                        (128)
\p{Block: UCAS}          \p{Block=
                        Unified_Canadian_Aboriginal_Syllabics}
                        (640)
\p{Block: UCAS_Ext}      \p{Block=
                        Unified_Canadian_Aboriginal_Syllabics_
                        Extended} (80)
\p{Block: Ugaritic}      (Single: \p{InUgaritic}; NOT \p{Ugaritic}
                        NOR \p{Is_Ugaritic}) (32)
\p{Block: Unified_Canadian_Aboriginal_Syllabics} (Short: \p{Blk=
                        UCAS}, \p{InUCAS}) (640)
\p{Block: Unified_Canadian_Aboriginal_Syllabics_Extended} (Short:
                        \p{Blk=UCASExt}, \p{InUCASExt}) (80)
\p{Block: Vai}           (Single: \p{InVai}; NOT \p{Vai} NOR
                        \p{Is_Vai}) (320)
\p{Block: Variation_Selectors} (Short: \p{Blk=VS}, \p{InVS}; NOT
                        \p{Variation_Selector} NOR \p{Is_VS})
                        (16)
\p{Block: Variation_Selectors_Supplement} (Short: \p{Blk=VSSup},
                        \p{InVSSup}) (240)
\p{Block: Vedic_Ext}     \p{Block=Vedic_Extensions} (48)
\p{Block: Vedic_Extensions} (Short: \p{Blk=VedicExt},
                        \p{InVedicExt}) (48)
\p{Block: Vertical_Forms} (Single: \p{InVerticalForms}) (16)
\p{Block: VS}            \p{Block=Variation_Selectors} (NOT
                        \p{Variation_Selector} NOR \p{Is_VS})
                        (16)
\p{Block: VS_Sup}        \p{Block=Variation_Selectors_Supplement}
                        (240)
\p{Block: Yi_Radicals}  (Single: \p{InYiRadicals}) (64)

```

	<code>\p{Block: Yi_Syllables}</code>	<code>(Single: \p{InYiSyllables})</code>	<code>(1168)</code>
	<code>\p{Block: Yijing}</code>	<code>\p{Block=Yijing_Hexagram_Symbols}</code>	<code>(64)</code>
	<code>\p{Block: Yijing_Hexagram_Symbols}</code>	<code>(Short: \p{Blk=Yijing},</code>	<code>\p{InYijing})</code>
X	<code>\p{Block_Elements}</code>	<code>\p{Block=Block_Elements}</code>	<code>(32)</code>
	<code>\p{Bopo}</code>	<code>\p{Bopomofo} (= \p{Script=Bopomofo})</code>	<code>(NOT</code>
		<code>\p{Block=Bopomofo})</code>	<code>(70)</code>
	<code>\p{Bopomofo}</code>	<code>\p{Script=Bopomofo} (Short: \p{Bopo};</code>	<code>NOT</code>
		<code>\p{Block=Bopomofo})</code>	<code>(70)</code>
X	<code>\p{Bopomofo_Ext}</code>	<code>\p{Bopomofo_Extended} (= \p{Block=</code>	<code>Bopomofo_Extended})</code>
		<code>(32)</code>	
X	<code>\p{Bopomofo_Extended}</code>	<code>\p{Block=Bopomofo_Extended} (Short:</code>	<code>\p{InBopomofoExt})</code>
		<code>(32)</code>	
X	<code>\p{Box_Drawing}</code>	<code>\p{Block=Box_Drawing}</code>	<code>(128)</code>
	<code>\p{Brah}</code>	<code>\p{Brahmi} (= \p{Script=Brahmi})</code>	<code>(NOT</code>
		<code>\p{Block=Brahmi})</code>	<code>(108)</code>
	<code>\p{Brahmi}</code>	<code>\p{Script=Brahmi} (Short: \p{Brah};</code>	<code>NOT</code>
		<code>\p{Block=Brahmi})</code>	<code>(108)</code>
	<code>\p{Brai}</code>	<code>\p{Braille} (= \p{Script=Braille})</code>	<code>(256)</code>
	<code>\p{Braille}</code>	<code>\p{Script=Braille} (Short: \p{Brai})</code>	<code>(256)</code>
X	<code>\p{Braille_Patterns}</code>	<code>\p{Block=Braille_Patterns} (Short:</code>	<code>\p{InBraille})</code>
		<code>(256)</code>	
	<code>\p{Bugi}</code>	<code>\p{Buginese} (= \p{Script=Buginese})</code>	<code>(NOT</code>
		<code>\p{Block=Buginese})</code>	<code>(30)</code>
	<code>\p{Buginese}</code>	<code>\p{Script=Buginese} (Short: \p{Bugi};</code>	<code>NOT</code>
		<code>\p{Block=Buginese})</code>	<code>(30)</code>
	<code>\p{Buhd}</code>	<code>\p{Buhid} (= \p{Script=Buhid})</code>	<code>(NOT</code>
		<code>\p{Block=Buhid})</code>	<code>(20)</code>
	<code>\p{Buhid}</code>	<code>\p{Script=Buhid} (Short: \p{Buhd};</code>	<code>NOT</code>
		<code>\p{Block=Buhid})</code>	<code>(20)</code>
X	<code>\p{Byzantine_Music}</code>	<code>\p{Byzantine_Musical_Symbols} (= \p{Block=</code>	<code>Byzantine_Musical_Symbols})</code>
		<code>(256)</code>	
X	<code>\p{Byzantine_Musical_Symbols}</code>	<code>\p{Block=Byzantine_Musical_Symbols}</code>	<code>(Short: \p{InByzantineMusic})</code>
		<code>(256)</code>	
	<code>\p{C}</code>	<code>\p{Other} (= \p{General_Category=Other})</code>	<code>(1_004_134)</code>
	<code>\p{Cakm}</code>	<code>\p{Chakma} (= \p{Script=Chakma})</code>	<code>(NOT</code>
		<code>\p{Block=Chakma})</code>	<code>(67)</code>
	<code>\p{Canadian_Aboriginal}</code>	<code>\p{Script=Canadian_Aboriginal} (Short:</code>	<code>\p{Cans})</code>
		<code>(710)</code>	
X	<code>\p{Canadian_Syllabics}</code>	<code>\p{Unified_Canadian_Aboriginal_Syllabics}</code>	<code>(= \p{Block=</code>
		<code>Unified_Canadian_Aboriginal_Syllabics})</code>	<code>(640)</code>
T	<code>\p{Canonical_Combining_Class: 0}</code>	<code>\p{Canonical_Combining_Class=</code>	<code>Not_Reordered}</code>
		<code>(1_113_459)</code>	
T	<code>\p{Canonical_Combining_Class: 1}</code>	<code>\p{Canonical_Combining_Class=</code>	<code>Overlay}</code>
		<code>(26)</code>	
T	<code>\p{Canonical_Combining_Class: 7}</code>	<code>\p{Canonical_Combining_Class=</code>	<code>Nukta}</code>
		<code>(13)</code>	
T	<code>\p{Canonical_Combining_Class: 8}</code>	<code>\p{Canonical_Combining_Class=</code>	<code>Kana_Voicing}</code>
		<code>(2)</code>	
T	<code>\p{Canonical_Combining_Class: 9}</code>	<code>\p{Canonical_Combining_Class=</code>	<code>Virama}</code>
		<code>(37)</code>	
T	<code>\p{Canonical_Combining_Class: 10}</code>	<code>\p{Canonical_Combining_Class=</code>	<code>CCC10}</code>
		<code>(1)</code>	

T \p{Canonical_Combining_Class: 11} \p{Canonical_Combining_Class=CCC11} (1)

T \p{Canonical_Combining_Class: 12} \p{Canonical_Combining_Class=CCC12} (1)

T \p{Canonical_Combining_Class: 13} \p{Canonical_Combining_Class=CCC13} (1)

T \p{Canonical_Combining_Class: 14} \p{Canonical_Combining_Class=CCC14} (1)

T \p{Canonical_Combining_Class: 15} \p{Canonical_Combining_Class=CCC15} (1)

T \p{Canonical_Combining_Class: 16} \p{Canonical_Combining_Class=CCC16} (1)

T \p{Canonical_Combining_Class: 17} \p{Canonical_Combining_Class=CCC17} (1)

T \p{Canonical_Combining_Class: 18} \p{Canonical_Combining_Class=CCC18} (2)

T \p{Canonical_Combining_Class: 19} \p{Canonical_Combining_Class=CCC19} (2)

T \p{Canonical_Combining_Class: 20} \p{Canonical_Combining_Class=CCC20} (1)

T \p{Canonical_Combining_Class: 21} \p{Canonical_Combining_Class=CCC21} (1)

T \p{Canonical_Combining_Class: 22} \p{Canonical_Combining_Class=CCC22} (1)

T \p{Canonical_Combining_Class: 23} \p{Canonical_Combining_Class=CCC23} (1)

T \p{Canonical_Combining_Class: 24} \p{Canonical_Combining_Class=CCC24} (1)

T \p{Canonical_Combining_Class: 25} \p{Canonical_Combining_Class=CCC25} (1)

T \p{Canonical_Combining_Class: 26} \p{Canonical_Combining_Class=CCC26} (1)

T \p{Canonical_Combining_Class: 27} \p{Canonical_Combining_Class=CCC27} (2)

T \p{Canonical_Combining_Class: 28} \p{Canonical_Combining_Class=CCC28} (2)

T \p{Canonical_Combining_Class: 29} \p{Canonical_Combining_Class=CCC29} (2)

T \p{Canonical_Combining_Class: 30} \p{Canonical_Combining_Class=CCC30} (2)

T \p{Canonical_Combining_Class: 31} \p{Canonical_Combining_Class=CCC31} (2)

T \p{Canonical_Combining_Class: 32} \p{Canonical_Combining_Class=CCC32} (2)

T \p{Canonical_Combining_Class: 33} \p{Canonical_Combining_Class=CCC33} (1)

T \p{Canonical_Combining_Class: 34} \p{Canonical_Combining_Class=CCC34} (1)

T \p{Canonical_Combining_Class: 35} \p{Canonical_Combining_Class=CCC35} (1)

T \p{Canonical_Combining_Class: 36} \p{Canonical_Combining_Class=CCC36} (1)

T \p{Canonical_Combining_Class: 84} \p{Canonical_Combining_Class=CCC84} (1)

T \p{Canonical_Combining_Class: 91} \p{Canonical_Combining_Class=CCC91} (1)

$\backslash p\{\text{Canonical_Combining_Class: 103}\} \backslash p\{\text{Canonical_Combining_Class=CCC103}\} (2)$
 $\backslash p\{\text{Canonical_Combining_Class: 107}\} \backslash p\{\text{Canonical_Combining_Class=CCC107}\} (4)$
 $\backslash p\{\text{Canonical_Combining_Class: 118}\} \backslash p\{\text{Canonical_Combining_Class=CCC118}\} (2)$
 $\backslash p\{\text{Canonical_Combining_Class: 122}\} \backslash p\{\text{Canonical_Combining_Class=CCC122}\} (4)$
 $\backslash p\{\text{Canonical_Combining_Class: 129}\} \backslash p\{\text{Canonical_Combining_Class=CCC129}\} (1)$
 $\backslash p\{\text{Canonical_Combining_Class: 130}\} \backslash p\{\text{Canonical_Combining_Class=CCC130}\} (6)$
 $\backslash p\{\text{Canonical_Combining_Class: 132}\} \backslash p\{\text{Canonical_Combining_Class=CCC132}\} (1)$
 $\backslash p\{\text{Canonical_Combining_Class: 133}\} \backslash p\{\text{Canonical_Combining_Class=CCC133}\} (0)$
 $\backslash p\{\text{Canonical_Combining_Class: 200}\} \backslash p\{\text{Canonical_Combining_Class=Attached_Below_Left}\} (0)$
 $\backslash p\{\text{Canonical_Combining_Class: 202}\} \backslash p\{\text{Canonical_Combining_Class=Attached_Below}\} (5)$
 $\backslash p\{\text{Canonical_Combining_Class: 214}\} \backslash p\{\text{Canonical_Combining_Class=Attached_Above}\} (1)$
 $\backslash p\{\text{Canonical_Combining_Class: 216}\} \backslash p\{\text{Canonical_Combining_Class=Attached_Above_Right}\} (9)$
 $\backslash p\{\text{Canonical_Combining_Class: 218}\} \backslash p\{\text{Canonical_Combining_Class=Below_Left}\} (1)$
 $\backslash p\{\text{Canonical_Combining_Class: 220}\} \backslash p\{\text{Canonical_Combining_Class=Below}\} (129)$
 $\backslash p\{\text{Canonical_Combining_Class: 222}\} \backslash p\{\text{Canonical_Combining_Class=Below_Right}\} (4)$
 $\backslash p\{\text{Canonical_Combining_Class: 224}\} \backslash p\{\text{Canonical_Combining_Class=Left}\} (2)$
 $\backslash p\{\text{Canonical_Combining_Class: 226}\} \backslash p\{\text{Canonical_Combining_Class=Right}\} (1)$
 $\backslash p\{\text{Canonical_Combining_Class: 228}\} \backslash p\{\text{Canonical_Combining_Class=Above_Left}\} (3)$
 $\backslash p\{\text{Canonical_Combining_Class: 230}\} \backslash p\{\text{Canonical_Combining_Class=Above}\} (349)$
 $\backslash p\{\text{Canonical_Combining_Class: 232}\} \backslash p\{\text{Canonical_Combining_Class=Above_Right}\} (4)$
 $\backslash p\{\text{Canonical_Combining_Class: 233}\} \backslash p\{\text{Canonical_Combining_Class=Double_Below}\} (4)$
 $\backslash p\{\text{Canonical_Combining_Class: 234}\} \backslash p\{\text{Canonical_Combining_Class=Double_Above}\} (5)$
 $\backslash p\{\text{Canonical_Combining_Class: 240}\} \backslash p\{\text{Canonical_Combining_Class=Iota_Subscript}\} (1)$
 $\backslash p\{\text{Canonical_Combining_Class: A}\} \backslash p\{\text{Canonical_Combining_Class=Above}\} (349)$
 $\backslash p\{\text{Canonical_Combining_Class: Above}\} (\text{Short: } \backslash p\{\text{Ccc=A}\}) (349)$
 $\backslash p\{\text{Canonical_Combining_Class: Above_Left}\} (\text{Short: } \backslash p\{\text{Ccc=AL}\}) (3)$
 $\backslash p\{\text{Canonical_Combining_Class: Above_Right}\} (\text{Short: } \backslash p\{\text{Ccc=AR}\}) (4)$
 $\backslash p\{\text{Canonical_Combining_Class: AL}\} \backslash p\{\text{Canonical_Combining_Class=Above_Left}\} (3)$
 $\backslash p\{\text{Canonical_Combining_Class: AR}\} \backslash p\{\text{Canonical_Combining_Class=Above_Right}\} (4)$
 $\backslash p\{\text{Canonical_Combining_Class: ATA}\} \backslash p\{\text{Canonical_Combining_Class=}$

```

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} (129)
\p{Canonical_Combining_Class: Below} (Short: \p{Ccc=B}) (129)
\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: CCC10} (Short: \p{Ccc=CCC10}) (1)
\p{Canonical_Combining_Class: CCC103} (Short: \p{Ccc=CCC103}) (2)
\p{Canonical_Combining_Class: CCC107} (Short: \p{Ccc=CCC107}) (4)
\p{Canonical_Combining_Class: CCC11} (Short: \p{Ccc=CCC11}) (1)
\p{Canonical_Combining_Class: CCC118} (Short: \p{Ccc=CCC118}) (2)
\p{Canonical_Combining_Class: CCC12} (Short: \p{Ccc=CCC12}) (1)
\p{Canonical_Combining_Class: CCC122} (Short: \p{Ccc=CCC122}) (4)
\p{Canonical_Combining_Class: CCC129} (Short: \p{Ccc=CCC129}) (1)
\p{Canonical_Combining_Class: CCC13} (Short: \p{Ccc=CCC13}) (1)
\p{Canonical_Combining_Class: CCC130} (Short: \p{Ccc=CCC130}) (6)
\p{Canonical_Combining_Class: CCC132} (Short: \p{Ccc=CCC132}) (1)
\p{Canonical_Combining_Class: CCC133} (Short: \p{Ccc=CCC133}) (0)
\p{Canonical_Combining_Class: CCC14} (Short: \p{Ccc=CCC14}) (1)
\p{Canonical_Combining_Class: CCC15} (Short: \p{Ccc=CCC15}) (1)
\p{Canonical_Combining_Class: CCC16} (Short: \p{Ccc=CCC16}) (1)
\p{Canonical_Combining_Class: CCC17} (Short: \p{Ccc=CCC17}) (1)
\p{Canonical_Combining_Class: CCC18} (Short: \p{Ccc=CCC18}) (2)
\p{Canonical_Combining_Class: CCC19} (Short: \p{Ccc=CCC19}) (2)
\p{Canonical_Combining_Class: CCC20} (Short: \p{Ccc=CCC20}) (1)
\p{Canonical_Combining_Class: CCC21} (Short: \p{Ccc=CCC21}) (1)
\p{Canonical_Combining_Class: CCC22} (Short: \p{Ccc=CCC22}) (1)
\p{Canonical_Combining_Class: CCC23} (Short: \p{Ccc=CCC23}) (1)
\p{Canonical_Combining_Class: CCC24} (Short: \p{Ccc=CCC24}) (1)
\p{Canonical_Combining_Class: CCC25} (Short: \p{Ccc=CCC25}) (1)
\p{Canonical_Combining_Class: CCC26} (Short: \p{Ccc=CCC26}) (1)
\p{Canonical_Combining_Class: CCC27} (Short: \p{Ccc=CCC27}) (2)
\p{Canonical_Combining_Class: CCC28} (Short: \p{Ccc=CCC28}) (2)
\p{Canonical_Combining_Class: CCC29} (Short: \p{Ccc=CCC29}) (2)
\p{Canonical_Combining_Class: CCC30} (Short: \p{Ccc=CCC30}) (2)
\p{Canonical_Combining_Class: CCC31} (Short: \p{Ccc=CCC31}) (2)
\p{Canonical_Combining_Class: CCC32} (Short: \p{Ccc=CCC32}) (2)
\p{Canonical_Combining_Class: CCC33} (Short: \p{Ccc=CCC33}) (1)

```

```

\p{Canonical_Combining_Class: CCC34} (Short: \p{Ccc=CCC34}) (1)
\p{Canonical_Combining_Class: CCC35} (Short: \p{Ccc=CCC35}) (1)
\p{Canonical_Combining_Class: CCC36} (Short: \p{Ccc=CCC36}) (1)
\p{Canonical_Combining_Class: CCC84} (Short: \p{Ccc=CCC84}) (1)
\p{Canonical_Combining_Class: CCC91} (Short: \p{Ccc=CCC91}) (1)
\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} (13)
\p{Canonical_Combining_Class: Not_Reordered} (Short: \p{Ccc=NR})
    (1_113_459)
\p{Canonical_Combining_Class: NR} \p{Canonical_Combining_Class=
    Not_Reordered} (1_113_459)
\p{Canonical_Combining_Class: Nukta} (Short: \p{Ccc=NK}) (13)
\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}) (37)
\p{Canonical_Combining_Class: VR} \p{Canonical_Combining_Class=
    Virama} (37)
\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}) (1799)
\p{Case_Ignorable: N*} (Short: \p{CI=N}, \p{CI}) (1_112_313)
\p{Case_Ignorable: Y*} (Short: \p{CI=Y}, \p{CI}) (1799)
\p{Cased} \p{Cased=Y} (3448)
\p{Cased: N*} (Single: \p{Cased}) (1_110_664)
\p{Cased: Y*} (Single: \p{Cased}) (3448)
\p{Cased_Letter} \p{General_Category=Cased_Letter} (Short:
    \p{LC}) (3223)
\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} (=

```

	<code>\p{Composition_Exclusion=Y}</code>) (81)
<code>\p{CE: *}</code>	<code>\p{Composition_Exclusion: *}</code>
<code>\p{Cf}</code>	<code>\p{Format}</code> (= <code>\p{General_Category=Format}</code>) (139)
<code>\p{Chakma}</code>	<code>\p{Script=Chakma}</code> (Short: <code>\p{Cakm}</code> ; NOT <code>\p{Block=Chakma}</code>) (67)
<code>\p{Cham}</code>	<code>\p{Script=Cham}</code> (NOT <code>\p{Block=Cham}</code>) (83)
<code>\p{Changes_When_Casefolded}</code>	<code>\p{Changes_When_Casefolded=Y}</code> (Short: <code>\p{CWCF}</code>) (1107)
<code>\p{Changes_When_Casefolded: N*}</code>	(Short: <code>\p{CWCF=N}</code> , <code>\p{CWCF}</code>) (1_113_005)
<code>\p{Changes_When_Casefolded: Y*}</code>	(Short: <code>\p{CWCF=Y}</code> , <code>\p{CWCF}</code>) (1107)
<code>\p{Changes_When_Casemapped}</code>	<code>\p{Changes_When_Casemapped=Y}</code> (Short: <code>\p{CWCM}</code>) (2138)
<code>\p{Changes_When_Casemapped: N*}</code>	(Short: <code>\p{CWCM=N}</code> , <code>\p{CWCM}</code>) (1_111_974)
<code>\p{Changes_When_Casemapped: Y*}</code>	(Short: <code>\p{CWCM=Y}</code> , <code>\p{CWCM}</code>) (2138)
<code>\p{Changes_When_Lowercased}</code>	<code>\p{Changes_When_Lowercased=Y}</code> (Short: <code>\p{CWL}</code>) (1043)
<code>\p{Changes_When_Lowercased: N*}</code>	(Short: <code>\p{CWL=N}</code> , <code>\p{CWL}</code>) (1_113_069)
<code>\p{Changes_When_Lowercased: Y*}</code>	(Short: <code>\p{CWL=Y}</code> , <code>\p{CWL}</code>) (1043)
<code>\p{Changes_When_NFKC_Casefolded}</code>	<code>\p{Changes_When_NFKC_Casefolded=Y}</code> (Short: <code>\p{CWKCF}</code>) (9944)
<code>\p{Changes_When_NFKC_Casefolded: N*}</code>	(Short: <code>\p{CWKCF=N}</code> , <code>\p{CWKCF}</code>) (1_104_168)
<code>\p{Changes_When_NFKC_Casefolded: Y*}</code>	(Short: <code>\p{CWKCF=Y}</code> , <code>\p{CWKCF}</code>) (9944)
<code>\p{Changes_When_Titlecased}</code>	<code>\p{Changes_When_Titlecased=Y}</code> (Short: <code>\p{CWT}</code>) (1099)
<code>\p{Changes_When_Titlecased: N*}</code>	(Short: <code>\p{CWT=N}</code> , <code>\p{CWT}</code>) (1_113_013)
<code>\p{Changes_When_Titlecased: Y*}</code>	(Short: <code>\p{CWT=Y}</code> , <code>\p{CWT}</code>) (1099)
<code>\p{Changes_When_Uppercased}</code>	<code>\p{Changes_When_Uppercased=Y}</code> (Short: <code>\p{CWU}</code>) (1126)
<code>\p{Changes_When_Uppercased: N*}</code>	(Short: <code>\p{CWU=N}</code> , <code>\p{CWU}</code>) (1_112_986)
<code>\p{Changes_When_Uppercased: Y*}</code>	(Short: <code>\p{CWU=Y}</code> , <code>\p{CWU}</code>) (1126)
<code>\p{Cher}</code>	<code>\p{Cherokee}</code> (= <code>\p{Script=Cherokee}</code>) (NOT <code>\p{Block=Cherokee}</code>) (85)
<code>\p{Cherokee}</code>	<code>\p{Script=Cherokee}</code> (Short: <code>\p{Cher}</code> ; NOT <code>\p{Block=Cherokee}</code>) (85)
<code>\p{CI}</code>	<code>\p{Case_Ignorable}</code> (= <code>\p{Case_Ignorable=Y}</code>) (1799)
<code>\p{CI: *}</code>	<code>\p{Case_Ignorable: *}</code>
X <code>\p{CJK}</code>	<code>\p{CJK_Unified_Ideographs}</code> (= <code>\p{Block=CJK_Unified_Ideographs}</code>) (20_992)
X <code>\p{CJK_Compat}</code>	<code>\p{CJK_Compatibility}</code> (= <code>\p{Block=CJK_Compatibility}</code>) (256)
X <code>\p{CJK_Compat_Forms}</code>	<code>\p{CJK_Compatibility_Forms}</code> (= <code>\p{Block=CJK_Compatibility_Forms}</code>) (32)
X <code>\p{CJK_Compat_Ideographs}</code>	<code>\p{CJK_Compatibility_Ideographs}</code> (= <code>\p{Block=CJK_Compatibility_Ideographs}</code>) (512)

- X `\p{CJK_Compat_Ideographs_Sup}`
`\p{CJK_Compatibility_Ideographs_Supplement}` (= `\p{Block=CJK_Compatibility_Ideographs_Supplement}`) (544)
- X `\p{CJK_Compatibility}` `\p{Block=CJK_Compatibility}` (Short: `\p{InCJKCompat}`) (256)
- X `\p{CJK_Compatibility_Forms}` `\p{Block=CJK_Compatibility_Forms}`
 (Short: `\p{InCJKCompatForms}`) (32)
- X `\p{CJK_Compatibility_Ideographs}` `\p{Block=CJK_Compatibility_Ideographs}` (Short: `\p{InCJKCompatIdeographs}`) (512)
- X `\p{CJK_Compatibility_Ideographs_Supplement}` `\p{Block=CJK_Compatibility_Ideographs_Supplement}`
 (Short: `\p{InCJKCompatIdeographsSup}`) (544)
- X `\p{CJK_Ext_A}` `\p{CJK_Unified_Ideographs_Extension_A}` (= `\p{Block=CJK_Unified_Ideographs_Extension_A}`) (6592)
- X `\p{CJK_Ext_B}` `\p{CJK_Unified_Ideographs_Extension_B}` (= `\p{Block=CJK_Unified_Ideographs_Extension_B}`) (42_720)
- X `\p{CJK_Ext_C}` `\p{CJK_Unified_Ideographs_Extension_C}` (= `\p{Block=CJK_Unified_Ideographs_Extension_C}`) (4160)
- X `\p{CJK_Ext_D}` `\p{CJK_Unified_Ideographs_Extension_D}` (= `\p{Block=CJK_Unified_Ideographs_Extension_D}`) (224)
- X `\p{CJK_Radicals_Sup}` `\p{CJK_Radicals_Supplement}` (= `\p{Block=CJK_Radicals_Supplement}`) (128)
- X `\p{CJK_Radicals_Supplement}` `\p{Block=CJK_Radicals_Supplement}`
 (Short: `\p{InCJKRadicalsSup}`) (128)
- X `\p{CJK_Strokes}` `\p{Block=CJK_Strokes}` (48)
- X `\p{CJK_Symbols}` `\p{CJK_Symbols_And_Punctuation}` (= `\p{Block=CJK_Symbols_And_Punctuation}`) (64)
- X `\p{CJK_Symbols_And_Punctuation}` `\p{Block=CJK_Symbols_And_Punctuation}` (Short: `\p{InCJKSymbols}`) (64)
- X `\p{CJK_Unified_Ideographs}` `\p{Block=CJK_Unified_Ideographs}`
 (Short: `\p{InCJK}`) (20_992)
- X `\p{CJK_Unified_Ideographs_Extension_A}` `\p{Block=CJK_Unified_Ideographs_Extension_A}`
 (Short: `\p{InCJKExtA}`) (6592)
- X `\p{CJK_Unified_Ideographs_Extension_B}` `\p{Block=CJK_Unified_Ideographs_Extension_B}`
 (Short: `\p{InCJKExtB}`) (42_720)
- X `\p{CJK_Unified_Ideographs_Extension_C}` `\p{Block=CJK_Unified_Ideographs_Extension_C}`
 (Short: `\p{InCJKExtC}`) (4160)
- X `\p{CJK_Unified_Ideographs_Extension_D}` `\p{Block=CJK_Unified_Ideographs_Extension_D}`

	(Short: <code>\p{InCJKEExtD}</code>) (224)
<code>\p{Close_Punctuation}</code>	<code>\p{General_Category=Close_Punctuation}</code> (Short: <code>\p{Pe}</code>) (71)
<code>\p{Cn}</code>	<code>\p{Unassigned}</code> (= <code>\p{General_Category=Unassigned}</code>) (864_414)
<code>\p{Cntrl}</code>	<code>\p{General_Category=Control}</code> Control characters (Short: <code>\p{Cc}</code>) (65)
<code>\p{Co}</code>	<code>\p{Private_Use}</code> (= <code>\p{General_Category=Private_Use}</code>) (NOT <code>\p{Private_Use_Area}</code>) (137_468)
X <code>\p{Combining_Diacritical_Marks}</code>	<code>\p{Block=Combining_Diacritical_Marks}</code> (Short: <code>\p{InDiacriticals}</code>) (112)
X <code>\p{Combining_Diacritical_Marks_For_Symbols}</code>	<code>\p{Block=Combining_Diacritical_Marks_For_Symbols}</code> (Short: <code>\p{InDiacriticalsForSymbols}</code>) (48)
X <code>\p{Combining_Diacritical_Marks_Supplement}</code>	<code>\p{Block=Combining_Diacritical_Marks_Supplement}</code> (Short: <code>\p{InDiacriticalsSup}</code>) (64)
X <code>\p{Combining_Half_Marks}</code>	<code>\p{Block=Combining_Half_Marks}</code> (Short: <code>\p{InHalfMarks}</code>) (16)
<code>\p{Combining_Mark}</code>	<code>\p{Mark}</code> (= <code>\p{General_Category=Mark}</code>) (1645)
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>) (6413)
X <code>\p{Common_Indic_Number_Forms}</code>	<code>\p{Block=Common_Indic_Number_Forms}</code> (Short: <code>\p{InIndicNumberForms}</code>) (16)
<code>\p{Comp_Ex}</code>	<code>\p{Full_Composition_Exclusion}</code> (= <code>\p{Full_Composition_Exclusion=Y}</code>) (1120)
<code>\p{Comp_Ex: *}</code>	<code>\p{Full_Composition_Exclusion: *}</code>
X <code>\p{Compat_Jamo}</code>	<code>\p{Hangul_Compatibility_Jamo}</code> (= <code>\p{Block=Hangul_Compatibility_Jamo}</code>) (96)
<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>) (137)
<code>\p{Coptic}</code>	<code>\p{Script=Coptic}</code> (Short: <code>\p{Copt}</code> ; NOT <code>\p{Block=Coptic}</code>) (137)
X <code>\p{Counting_Rod}</code>	<code>\p{Counting_Rod_Numerals}</code> (= <code>\p{Block=Counting_Rod_Numerals}</code>) (32)
X <code>\p{Counting_Rod_Numerals}</code>	<code>\p{Block=Counting_Rod_Numerals}</code> (Short: <code>\p{InCountingRod}</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}</code>	<code>\p{Cuneiform_Numbers_And_Punctuation}</code> (= <code>\p{Block=Cuneiform_Numbers_And_Punctuation}</code>) (128)
X <code>\p{Cuneiform_Numbers_And_Punctuation}</code>	<code>\p{Block=Cuneiform_Numbers_And_Punctuation}</code> (Short: <code>\p{InCuneiformNumbers}</code>) (128)
<code>\p{Currency_Symbol}</code>	<code>\p{General_Category=Currency_Symbol}</code> (Short: <code>\p{Sc}</code>) (49)
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>) (1107)
<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>) (2138)
<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>) (9944)
<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>) (1043)
<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>) (1099)
<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>) (1126)
<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>) (417)
X <code>\p{Cyrillic_Ext_A}</code>	<code>\p{Cyrillic_Extended_A}</code> (= <code>\p{Block=Cyrillic_Extended_A}</code>) (32)
X <code>\p{Cyrillic_Ext_B}</code>	<code>\p{Cyrillic_Extended_B}</code> (= <code>\p{Block=Cyrillic_Extended_B}</code>) (96)
X <code>\p{Cyrillic_Extended_A}</code>	<code>\p{Block=Cyrillic_Extended_A}</code> (Short: <code>\p{InCyrillicExtA}</code>) (32)
X <code>\p{Cyrillic_Extended_B}</code>	<code>\p{Block=Cyrillic_Extended_B}</code> (Short: <code>\p{InCyrillicExtB}</code>) (96)
X <code>\p{Cyrillic_Sup}</code>	<code>\p{Cyrillic_Supplement}</code> (= <code>\p{Block=Cyrillic_Supplement}</code>) (48)
X <code>\p{Cyrillic_Supplement}</code>	<code>\p{Block=Cyrillic_Supplement}</code> (Short: <code>\p{InCyrillicSup}</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>) (417)
<code>\p{Dash}</code>	<code>\p{Dash=Y}</code> (27)
<code>\p{Dash: N*}</code>	(Single: <code>\p{Dash}</code>) (1_114_085)
<code>\p{Dash: Y*}</code>	(Single: <code>\p{Dash}</code>) (27)
<code>\p{Dash_Punctuation}</code>	<code>\p{General_Category=Dash_Punctuation}</code>

(Short: `\p{Pd}`) (23)

`\p{Decimal_Number}` `\p{Digit}` (= `\p{General_Category=Decimal_Number}`) (460)

`\p{Decomposition_Type: Can}` `\p{Decomposition_Type=Canonical}` (13_225)

`\p{Decomposition_Type: Canonical}` (Short: `\p{Dt=Can}`) (13_225)

`\p{Decomposition_Type: Circle}` (Short: `\p{Dt=Enc}`) (240)

`\p{Decomposition_Type: Com}` `\p{Decomposition_Type=Compat}` (720)

`\p{Decomposition_Type: Compat}` (Short: `\p{Dt=Com}`) (720)

`\p{Decomposition_Type: Enc}` `\p{Decomposition_Type=Circle}` (240)

`\p{Decomposition_Type: Fin}` `\p{Decomposition_Type=Final}` (240)

`\p{Decomposition_Type: Final}` (Short: `\p{Dt=Fin}`) (240)

`\p{Decomposition_Type: Font}` (Short: `\p{Dt=Font}`) (1184)

`\p{Decomposition_Type: Fra}` `\p{Decomposition_Type=Fraction}` (20)

`\p{Decomposition_Type: Fraction}` (Short: `\p{Dt=Fra}`) (20)

`\p{Decomposition_Type: Init}` `\p{Decomposition_Type=Initial}` (171)

`\p{Decomposition_Type: Initial}` (Short: `\p{Dt=Init}`) (171)

`\p{Decomposition_Type: Iso}` `\p{Decomposition_Type=Isolated}` (238)

`\p{Decomposition_Type: Isolated}` (Short: `\p{Dt=Iso}`) (238)

`\p{Decomposition_Type: Med}` `\p{Decomposition_Type=Medial}` (82)

`\p{Decomposition_Type: Medial}` (Short: `\p{Dt=Med}`) (82)

`\p{Decomposition_Type: Nar}` `\p{Decomposition_Type=Narrow}` (122)

`\p{Decomposition_Type: Narrow}` (Short: `\p{Dt=Nar}`) (122)

`\p{Decomposition_Type: Nb}` `\p{Decomposition_Type=Nobreak}` (5)

`\p{Decomposition_Type: Nobreak}` (Short: `\p{Dt=Nb}`) (5)

`\p{Decomposition_Type: Non_Canon}` `\p{Decomposition_Type=Non_Canonical}` (Perl extension) (3655)

`\p{Decomposition_Type: Non_Canonical}` Union of all non-canonical decompositions (Short: `\p{Dt=NonCanon}`) (Perl extension) (3655)

`\p{Decomposition_Type: None}` (Short: `\p{Dt=None}`) (1_097_232)

`\p{Decomposition_Type: Small}` (Short: `\p{Dt=Sml}`) (26)

`\p{Decomposition_Type: Sml}` `\p{Decomposition_Type=Small}` (26)

`\p{Decomposition_Type: Sqr}` `\p{Decomposition_Type=Square}` (284)

`\p{Decomposition_Type: Square}` (Short: `\p{Dt=Sqr}`) (284)

`\p{Decomposition_Type: Sub}` (Short: `\p{Dt=Sub}`) (38)

`\p{Decomposition_Type: Sup}` `\p{Decomposition_Type=Super}` (146)

`\p{Decomposition_Type: Super}` (Short: `\p{Dt=Sup}`) (146)

`\p{Decomposition_Type: Vert}` `\p{Decomposition_Type=Vertical}` (35)

`\p{Decomposition_Type: Vertical}` (Short: `\p{Dt=Vert}`) (35)

`\p{Decomposition_Type: Wide}` (Short: `\p{Dt=Wide}`) (104)

`\p{Default_Ignorable_Code_Point}` `\p{Default_Ignorable_Code_Point=Y}` (Short: `\p{DI}`) (4167)

`\p{Default_Ignorable_Code_Point: N*}` (Short: `\p{DI=N}`, `\P{DI}`) (1_109_945)

`\p{Default_Ignorable_Code_Point: Y*}` (Short: `\p{DI=Y}`, `\P{DI}`) (4167)

`\p{Dep}` `\p{Deprecated}` (= `\p{Deprecated=Y}`) (111)

`\p{Dep: *}` `\p{Deprecated: *}`

`\p{Deprecated}` `\p{Deprecated=Y}` (Short: `\p{Dep}`) (111)

`\p{Deprecated: N*}` (Short: `\p{Dep=N}`, `\P{Dep}`) (1_114_001)

`\p{Deprecated: Y*}` (Short: `\p{Dep=Y}`, `\P{Dep}`) (111)

`\p{Deseret}` `\p{Script=Deseret}` (Short: `\p{Dsrt}`) (80)

`\p{Deva}` `\p{Devanagari}` (= `\p{Script=Devanagari}`) (NOT `\p{Block=Devanagari}`) (151)

`\p{Devanagari}` `\p{Script=Devanagari}` (Short: `\p{Deva}`;

	NOT <code>\p{Block=Devanagari}</code>) (151)
X <code>\p{Devanagari_Ext}</code>	<code>\p{Devanagari_Extended}</code> (= <code>\p{Block=Devanagari_Extended}</code>) (32)
X <code>\p{Devanagari_Extended}</code>	<code>\p{Block=Devanagari_Extended}</code> (Short: <code>\p{InDevanagariExt}</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>) (693)
<code>\p{Dia: *}</code>	<code>\p{Diacritic: *}</code>
<code>\p{Diacritic}</code>	<code>\p{Diacritic=Y}</code> (Short: <code>\p{Dia}</code>) (693)
<code>\p{Diacritic: N*}</code>	(Short: <code>\p{Dia=N}</code> , <code>\p{Dia}</code>) (1_113_419)
<code>\p{Diacritic: Y*}</code>	(Short: <code>\p{Dia=Y}</code> , <code>\p{Dia}</code>) (693)
X <code>\p{Diacriticals}</code>	<code>\p{Combining_Diacritical_Marks}</code> (= <code>\p{Block=Combining_Diacritical_Marks}</code>) (112)
X <code>\p{Diacriticals_For_Symbols}</code>	<code>\p{Combining_Diacritical_Marks_For_Symbols}</code> (= <code>\p{Block=Combining_Diacritical_Marks_For_Symbols}</code>) (48)
X <code>\p{Diacriticals_Sup}</code>	<code>\p{Combining_Diacritical_Marks_Supplement}</code> (= <code>\p{Block=Combining_Diacritical_Marks_Supplement}</code>) (64)
<code>\p{Digit}</code>	<code>\p{General_Category=Decimal_Number}</code> [0-9] + all other decimal digits (Short: <code>\p{Nd}</code>) (460)
X <code>\p{Dingbats}</code>	<code>\p{Block=Dingbats}</code> (192)
X <code>\p{Domino}</code>	<code>\p{Domino_Tiles}</code> (= <code>\p{Block=Domino_Tiles}</code>) (112)
X <code>\p{Domino_Tiles}</code>	<code>\p{Block=Domino_Tiles}</code> (Short: <code>\p{InDomino}</code>) (112)
<code>\p{Dsrt}</code>	<code>\p{Deseret}</code> (= <code>\p{Script=Deseret}</code>) (80)
<code>\p{Dt: *}</code>	<code>\p{Decomposition_Type: *}</code>
<code>\p{Ea: *}</code>	<code>\p{East_Asian_Width: *}</code>
<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_894)
<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_894)
<code>\p{East_Asian_Width: W}</code>	<code>\p{East_Asian_Width=Wide}</code> (173_134)
<code>\p{East_Asian_Width: Wide}</code>	(Short: <code>\p{Ea=W}</code>) (173_134)
<code>\p{Egyp}</code>	<code>\p{Egyptian_Hieroglyphs}</code> (= <code>\p{Script=Egyptian_Hieroglyphs}</code>) (NOT <code>\p{Block=Egyptian_Hieroglyphs}</code>) (1071)
<code>\p{Egyptian_Hieroglyphs}</code>	<code>\p{Script=Egyptian_Hieroglyphs}</code> (Short: <code>\p{Egyp}</code> ; NOT <code>\p{Block=Egyptian_Hieroglyphs}</code>) (1071)
X <code>\p{Emoticons}</code>	<code>\p{Block=Emoticons}</code> (80)

X	<code>\p{Enclosed_Alphanum}</code>	<code>\p{Enclosed_Alphanumerics}</code> (= <code>\p{Block=Enclosed_Alphanumerics}</code>) (160)
X	<code>\p{Enclosed_Alphanum_Sup}</code>	<code>\p{Enclosed_Alphanumeric_Supplement}</code> (= <code>\p{Block=Enclosed_Alphanumeric_Supplement}</code>) (256)
X	<code>\p{Enclosed_Alphanumeric_Supplement}</code>	<code>\p{Block=Enclosed_Alphanumeric_Supplement}</code> (Short: <code>\p{InEnclosedAlphanumSup}</code>) (256)
X	<code>\p{Enclosed_Alphanumerics}</code>	<code>\p{Block=Enclosed_Alphanumerics}</code> (Short: <code>\p{InEnclosedAlphanum}</code>) (160)
X	<code>\p{Enclosed_CJK}</code>	<code>\p{Enclosed_CJK_Letters_And_Months}</code> (= <code>\p{Block=Enclosed_CJK_Letters_And_Months}</code>) (256)
X	<code>\p{Enclosed_CJK_Letters_And_Months}</code>	<code>\p{Block=Enclosed_CJK_Letters_And_Months}</code> (Short: <code>\p{InEnclosedCJK}</code>) (256)
X	<code>\p{Enclosed_Ideographic_Sup}</code>	<code>\p{Enclosed_Ideographic_Supplement}</code> (= <code>\p{Block=Enclosed_Ideographic_Supplement}</code>) (256)
X	<code>\p{Enclosed_Ideographic_Supplement}</code>	<code>\p{Block=Enclosed_Ideographic_Supplement}</code> (Short: <code>\p{InEnclosedIdeographicSup}</code>) (256)
	<code>\p{Enclosing_Mark}</code>	<code>\p{General_Category=Enclosing_Mark}</code> (Short: <code>\p{Me}</code>) (12)
	<code>\p{Ethi}</code>	<code>\p{Ethiopic}</code> (= <code>\p{Script=Ethiopic}</code>) (NOT <code>\p{Block=Ethiopic}</code>) (495)
	<code>\p{Ethiopic}</code>	<code>\p{Script=Ethiopic}</code> (Short: <code>\p{Ethi}</code> ; NOT <code>\p{Block=Ethiopic}</code>) (495)
X	<code>\p{Ethiopic_Ext}</code>	<code>\p{Ethiopic_Extended}</code> (= <code>\p{Block=Ethiopic_Extended}</code>) (96)
X	<code>\p{Ethiopic_Ext_A}</code>	<code>\p{Ethiopic_Extended_A}</code> (= <code>\p{Block=Ethiopic_Extended_A}</code>) (48)
X	<code>\p{Ethiopic_Extended}</code>	<code>\p{Block=Ethiopic_Extended}</code> (Short: <code>\p{InEthiopicExt}</code>) (96)
X	<code>\p{Ethiopic_Extended_A}</code>	<code>\p{Block=Ethiopic_Extended_A}</code> (Short: <code>\p{InEthiopicExtA}</code>) (48)
X	<code>\p{Ethiopic_Sup}</code>	<code>\p{Ethiopic_Supplement}</code> (= <code>\p{Block=Ethiopic_Supplement}</code>) (32)
X	<code>\p{Ethiopic_Supplement}</code>	<code>\p{Block=Ethiopic_Supplement}</code> (Short: <code>\p{InEthiopicSup}</code>) (32)
	<code>\p{Ext}</code>	<code>\p{Extender}</code> (= <code>\p{Extender=Y}</code>) (31)
	<code>\p{Ext: *}</code>	<code>\p{Extender: *}</code>
	<code>\p{Extender}</code>	<code>\p{Extender=Y}</code> (Short: <code>\p{Ext}</code>) (31)
	<code>\p{Extender: N*}</code>	(Short: <code>\p{Ext=N}</code> , <code>\p{Ext}</code>) (1_114_081)
	<code>\p{Extender: Y*}</code>	(Short: <code>\p{Ext=Y}</code> , <code>\p{Ext}</code>) (31)
	<code>\p{Final_Punctuation}</code>	<code>\p{General_Category=Final_Punctuation}</code> (Short: <code>\p{Pf}</code>) (10)
	<code>\p{Format}</code>	<code>\p{General_Category=Format}</code> (Short: <code>\p{Cf}</code>) (139)
	<code>\p{Full_Composition_Exclusion}</code>	<code>\p{Full_Composition_Exclusion=Y}</code> (Short: <code>\p{CompEx}</code>) (1120)
	<code>\p{Full_Composition_Exclusion: N*}</code>	(Short: <code>\p{CompEx=N}</code> , <code>\p{CompEx}</code>) (1_112_992)
	<code>\p{Full_Composition_Exclusion: Y*}</code>	(Short: <code>\p{CompEx=Y}</code> , <code>\p{CompEx}</code>) (1120)
	<code>\p{Gc: *}</code>	<code>\p{General_Category: *}</code>

```

\p{GCB: *} \p{Grapheme_Cluster_Break: *}
\p{General_Category: C} \p{General_Category=Other} (1_004_134)
\p{General_Category: Cased_Letter} [\p{Ll}\p{Lu}\p{Lt}] (Short:
    \p{Gc=LC}, \p{LC}) (3223)
\p{General_Category: Cc} \p{General_Category=Control} (65)
\p{General_Category: Cf} \p{General_Category=Format} (139)
\p{General_Category: Close_Punctuation} (Short: \p{Gc=Pe}, \p{Pe})
    (71)
\p{General_Category: Cn} \p{General_Category=Unassigned} (864_414)
\p{General_Category: Cntrl} \p{General_Category=Control} (65)
\p{General_Category: Co} \p{General_Category=Private_Use} (137_468)
\p{General_Category: Combining_Mark} \p{General_Category=Mark}
    (1645)
\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})
    (49)
\p{General_Category: Dash_Punctuation} (Short: \p{Gc=Pd}, \p{Pd})
    (23)
\p{General_Category: Decimal_Number} (Short: \p{Gc=Nd}, \p{Nd})
    (460)
\p{General_Category: Digit} \p{General_Category=Decimal_Number}
    (460)
\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}) (139)
\p{General_Category: Initial_Punctuation} (Short: \p{Gc=Pi},
    \p{Pi}) (12)
\p{General_Category: L} \p{General_Category=Letter} (101_013)
X \p{General_Category: L&} \p{General_Category=Cased_Letter} (3223)
X \p{General_Category: L_} \p{General_Category=Cased_Letter} Note
    the trailing '_' matters in spite of
    loose matching rules. (3223)
\p{General_Category: LC} \p{General_Category=Cased_Letter} (3223)
\p{General_Category: Letter} (Short: \p{Gc=L}, \p{L}) (101_013)
\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)
    (1751)
\p{General_Category: Lm} \p{General_Category=Modifier_Letter} (237)
\p{General_Category: Lo} \p{General_Category=Other_Letter} (97_553)
\p{General_Category: Lowercase_Letter} (Short: \p{Gc=Ll}, \p{Ll};
    /i= General_Category=Cased_Letter) (1751)
\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)
    (1441)
\p{General_Category: M} \p{General_Category=Mark} (1645)
\p{General_Category: Mark} (Short: \p{Gc=M}, \p{M}) (1645)
    
```

```

\p{General_Category: Math_Symbol} (Short: \p{Gc=Sm}, \p{Sm}) (952)
\p{General_Category: Mc} \p{General_Category=Spacing_Mark} (353)
\p{General_Category: Me} \p{General_Category=Enclosing_Mark} (12)
\p{General_Category: Mn} \p{General_Category=Nonspacing_Mark}
(1280)
\p{General_Category: Modifier_Letter} (Short: \p{Gc=Lm}, \p{Lm})
(237)
\p{General_Category: Modifier_Symbol} (Short: \p{Gc=Sk}, \p{Sk})
(115)
\p{General_Category: N} \p{General_Category=Number} (1148)
\p{General_Category: Nd} \p{General_Category=Decimal_Number} (460)
\p{General_Category: Nl} \p{General_Category=Letter_Number} (224)
\p{General_Category: No} \p{General_Category=Other_Number} (464)
\p{General_Category: Nonspacing_Mark} (Short: \p{Gc=Mn}, \p{Mn})
(1280)
\p{General_Category: Number} (Short: \p{Gc=N}, \p{N}) (1148)
\p{General_Category: Open_Punctuation} (Short: \p{Gc=Ps}, \p{Ps})
(72)
\p{General_Category: Other} (Short: \p{Gc=C}, \p{C}) (1_004_134)
\p{General_Category: Other_Letter} (Short: \p{Gc=Lo}, \p{Lo})
(97_553)
\p{General_Category: Other_Number} (Short: \p{Gc=No}, \p{No}) (464)
\p{General_Category: Other_Punctuation} (Short: \p{Gc=Po}, \p{Po})
(434)
\p{General_Category: Other_Symbol} (Short: \p{Gc=So}, \p{So})
(4404)
\p{General_Category: P} \p{General_Category=Punctuation} (632)
\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} (23)
\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}
(434)
\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} (632)
\p{General_Category: Punctuation} (Short: \p{Gc=P}, \p{P}) (632)
\p{General_Category: S} \p{General_Category=Symbol} (5520)
\p{General_Category: Sc} \p{General_Category=Currency_Symbol} (49)
\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} (952)
\p{General_Category: So} \p{General_Category=Other_Symbol} (4404)
\p{General_Category: Space_Separator} (Short: \p{Gc=Zs}, \p{Zs})
(18)
\p{General_Category: Spacing_Mark} (Short: \p{Gc=Mc}, \p{Mc}) (353)
\p{General_Category: Surrogate} (Short: \p{Gc=Cs}, \p{Cs}) (2048)
\p{General_Category: Symbol} (Short: \p{Gc=S}, \p{S}) (5520)

```

```

\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})
    (864_414)
\p{General_Category: Uppercase_Letter} (Short: \p{Gc=Lu}, \p{Lu};
    /i= General_Category=Cased_Letter) (1441)
\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} (Short:
    \p{InPunctuation}) (112)
X \p{Geometric_Shapes} \p{Block=Geometric_Shapes} (96)
\p{Geor} \p{Georgian} (= \p{Script=Georgian}) (NOT
    \p{Block=Georgian}) (127)
\p{Georgian} \p{Script=Georgian} (Short: \p{Geor}; NOT
    \p{Block=Georgian}) (127)
X \p{Georgian_Sup} \p{Georgian_Supplement} (= \p{Block=
    Georgian_Supplement}) (48)
X \p{Georgian_Supplement} \p{Block=Georgian_Supplement} (Short:
    \p{InGeorgianSup}) (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)
\p{Goth} \p{Gothic} (= \p{Script=Gothic}) (NOT
    \p{Block=Gothic}) (27)
\p{Gothic} \p{Script=Gothic} (Short: \p{Goth}; NOT
    \p{Block=Gothic}) (27)
\p{Gr_Base} \p{Grapheme_Base} (= \p{Grapheme_Base=Y})
    (108_661)
\p{Gr_Base: *} \p{Grapheme_Base: *}
\p{Gr_Ext} \p{Grapheme_Extend} (= \p{Grapheme_Extend=
    Y}) (1317)
\p{Gr_Ext: *} \p{Grapheme_Extend: *}
\p{Graph} Characters that are graphical (247_565)
\p{Grapheme_Base} \p{Grapheme_Base=Y} (Short: \p{GrBase})
    (108_661)
\p{Grapheme_Base: N*} (Short: \p{GrBase=N}, \p{GrBase})
    (1_005_451)
\p{Grapheme_Base: Y*} (Short: \p{GrBase=Y}, \p{GrBase}) (108_661)
\p{Grapheme_Cluster_Break: CN} \p{Grapheme_Cluster_Break=Control}
    (6023)
\p{Grapheme_Cluster_Break: Control} (Short: \p{GCB=CN}) (6023)
\p{Grapheme_Cluster_Break: CR} (Short: \p{GCB=CR}) (1)
\p{Grapheme_Cluster_Break: EX} \p{Grapheme_Cluster_Break=Extend}
    (1317)
\p{Grapheme_Cluster_Break: Extend} (Short: \p{GCB=EX}) (1317)
\p{Grapheme_Cluster_Break: L} (Short: \p{GCB=L}) (125)
\p{Grapheme_Cluster_Break: LF} (Short: \p{GCB=LF}) (1)
\p{Grapheme_Cluster_Break: LV} (Short: \p{GCB=LV}) (399)
\p{Grapheme_Cluster_Break: LVT} (Short: \p{GCB=LVT}) (10_773)
\p{Grapheme_Cluster_Break: Other} (Short: \p{GCB=XX}) (1_094_924)
\p{Grapheme_Cluster_Break: PP} \p{Grapheme_Cluster_Break=Prepend}
    (0)
    
```

```

\p{Grapheme_Cluster_Break: Prepend} (Short: \p{GCB=PP}) (0)
\p{Grapheme_Cluster_Break: Regional_Indicator} (Short: \p{GCB=RI})
(26)
\p{Grapheme_Cluster_Break: RI} \p{Grapheme_Cluster_Break=
Regional_Indicator} (26)
\p{Grapheme_Cluster_Break: SM} \p{Grapheme_Cluster_Break=
SpacingMark} (291)
\p{Grapheme_Cluster_Break: SpacingMark} (Short: \p{GCB=SM}) (291)
\p{Grapheme_Cluster_Break: T} (Short: \p{GCB=T}) (137)
\p{Grapheme_Cluster_Break: V} (Short: \p{GCB=V}) (95)
\p{Grapheme_Cluster_Break: XX} \p{Grapheme_Cluster_Break=Other}
(1_094_924)
\p{Grapheme_Extend} \p{Grapheme_Extend=Y} (Short: \p{GrExt})
(1317)
\p{Grapheme_Extend: N*} (Short: \p{GrExt=N}, \p{GrExt}) (1_112_795)
\p{Grapheme_Extend: Y*} (Short: \p{GrExt=Y}, \p{GrExt}) (1317)
\p{Greek} \p{Script=Greek} (Short: \p{Grek}; NOT
\p{Greek_And_Coptic}) (511)
X \p{Greek_And_Coptic} \p{Block=Greek_And_Coptic} (Short:
\p{InGreek}) (144)
X \p{Greek_Ext} \p{Greek_Extended} (= \p{Block=
Greek_Extended}) (256)
X \p{Greek_Extended} \p{Block=Greek_Extended} (Short:
\p{InGreekExt}) (256)
\p{Grek} \p{Greek} (= \p{Script=Greek}) (NOT
\p{Greek_And_Coptic}) (511)
\p{Gujarati} \p{Script=Gujarati} (Short: \p{Gujr}; NOT
\p{Block=Gujarati}) (84)
\p{Gujr} \p{Gujarati} (= \p{Script=Gujarati}) (NOT
\p{Block=Gujarati}) (84)
\p{Gurmukhi} \p{Script=Gurmukhi} (Short: \p{Guru}; NOT
\p{Block=Gurmukhi}) (79)
\p{Guru} \p{Gurmukhi} (= \p{Script=Gurmukhi}) (NOT
\p{Block=Gurmukhi}) (79)
X \p{Half_And_Full_Forms} \p{Halfwidth_And_Fullwidth_Forms} (=
\p{Block=Halfwidth_And_Fullwidth_Forms})
(240)
X \p{Half_Marks} \p{Combining_Half_Marks} (= \p{Block=
Combining_Half_Marks}) (16)
X \p{Halfwidth_And_Fullwidth_Forms} \p{Block=
Halfwidth_And_Fullwidth_Forms} (Short:
\p{InHalfAndFullForms}) (240)
\p{Han} \p{Script=Han} (75_963)
\p{Hang} \p{Hangul} (= \p{Script=Hangul}) (NOT
\p{Hangul_Syllables}) (11_739)
\p{Hangul} \p{Script=Hangul} (Short: \p{Hang}; NOT
\p{Hangul_Syllables}) (11_739)
X \p{Hangul_Compatibility_Jamo} \p{Block=Hangul_Compatibility_Jamo}
(Short: \p{InCompatJamo}) (96)
X \p{Hangul_Jamo} \p{Block=Hangul_Jamo} (Short: \p{InJamo})
(256)
X \p{Hangul_Jamo_Extended_A} \p{Block=Hangul_Jamo_Extended_A}
(Short: \p{InJamoExtA}) (32)
X \p{Hangul_Jamo_Extended_B} \p{Block=Hangul_Jamo_Extended_B}
(Short: \p{InJamoExtB}) (80)
\p{Hangul_Syllable_Type: L} \p{Hangul_Syllable_Type=Leading_Jamo}

```

	(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=</code> <code>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=</code> <code>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> (Short: <code>\p{InHangul}</code>) (11_184)
<code>\p{Hani}</code>	<code>\p{Han}</code> (= <code>\p{Script=Han}</code>) (75_963)
<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=</code> <code>High_Private_Use_Surrogates}</code> (Short: <code>\p{InHighPUSurrogates}</code>) (128)
X <code>\p{High_PU_Surrogates}</code>	<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> ; NOT

	<code>\p{Ideographic_Description_Characters}</code>	(103_355)
<code>\p{ID_Continue: N*}</code>	(Short: <code>\p{IDC=N}</code> , <code>\P{IDC}</code>)	(1_010_757)
<code>\p{ID_Continue: Y*}</code>	(Short: <code>\p{IDC=Y}</code> , <code>\P{IDC}</code>)	(103_355)
<code>\p{ID_Start}</code>	<code>\p{ID_Start=Y}</code> (Short: <code>\p{IDS}</code>)	(101_240)
<code>\p{ID_Start: N*}</code>	(Short: <code>\p{IDS=N}</code> , <code>\P{IDS}</code>)	(1_012_872)
<code>\p{ID_Start: Y*}</code>	(Short: <code>\p{IDS=Y}</code> , <code>\P{IDS}</code>)	(101_240)
<code>\p{IDC}</code>	<code>\p{ID_Continue}</code> (= <code>\p{ID_Continue=Y}</code>) (NOT <code>\p{Ideographic_Description_Characters}</code>)	(103_355)
<code>\p{IDC: *}</code>	<code>\p{ID_Continue: *}</code>	
<code>\p{Ideo}</code>	<code>\p{Ideographic}</code> (= <code>\p{Ideographic=Y}</code>)	(75_633)
<code>\p{Ideo: *}</code>	<code>\p{Ideographic: *}</code>	
<code>\p{Ideographic}</code>	<code>\p{Ideographic=Y}</code> (Short: <code>\p{Ideo}</code>)	(75_633)
<code>\p{Ideographic: N*}</code>	(Short: <code>\p{Ideo=N}</code> , <code>\P{Ideo}</code>)	(1_038_479)
<code>\p{Ideographic: Y*}</code>	(Short: <code>\p{Ideo=Y}</code> , <code>\P{Ideo}</code>)	(75_633)
X <code>\p{Ideographic_Description_Characters}</code>	<code>\p{Block=</code> <code>Ideographic_Description_Characters}</code>	(Short: <code>\p{InIDC}</code>) (16)
<code>\p{IDS}</code>	<code>\p{ID_Start}</code> (= <code>\p{ID_Start=Y}</code>)	(101_240)
<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>	
X <code>\p{Indic_Number_Forms}</code>	<code>\p{Common_Indic_Number_Forms}</code> (= <code>\p{Block=</code> <code>Common_Indic_Number_Forms}</code>)	(16)
<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=</code> <code>Inscriptional_Parthian}</code>)	(30)

X	<code>\p{IPA_Ext}</code>	<code>\p{IPA_Extensions}</code> (= <code>\p{Block=IPA_Extensions}</code>) (96)
X	<code>\p{IPA_Extensions}</code>	<code>\p{Block=IPA_Extensions}</code> (Short: <code>\p{InIPAExt}</code>) (96)
	<code>\p{Is_*}</code>	<code>\p{*}</code> (Any exceptions are individually noted beginning with the word NOT.) If an entry has flag(s) at its beginning, like "D", the "Is_" form has the same flag(s)
	<code>\p{Ital}</code>	<code>\p{Old_Italic}</code> (= <code>\p{Script=Old_Italic}</code>) (NOT <code>\p{Block=Old_Italic}</code>) (35)
X	<code>\p{Jamo}</code>	<code>\p{Hangul_Jamo}</code> (= <code>\p{Block=Hangul_Jamo}</code>) (256)
X	<code>\p{Jamo_Ext_A}</code>	<code>\p{Hangul_Jamo_Extended_A}</code> (= <code>\p{Block=Hangul_Jamo_Extended_A}</code>) (32)
X	<code>\p{Jamo_Ext_B}</code>	<code>\p{Hangul_Jamo_Extended_B}</code> (= <code>\p{Block=Hangul_Jamo_Extended_B}</code>) (80)
	<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>) (20)
	<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>) (10)
	<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>) (18)
	<code>\p{Joining_Group: Hamza_On_Heh_Goal}</code>	(Short: <code>\p{Jg=HamzaOnHehGoal}</code>) (1)
	<code>\p{Joining_Group: He}</code>	(Short: <code>\p{Jg=He}</code>) (1)
	<code>\p{Joining_Group: Heh}</code>	(Short: <code>\p{Jg=Heh}</code>) (1)
	<code>\p{Joining_Group: Heh_Goal}</code>	(Short: <code>\p{Jg=HehGoal}</code>) (2)
	<code>\p{Joining_Group: Heth}</code>	(Short: <code>\p{Jg=Heth}</code>) (1)
	<code>\p{Joining_Group: Kaf}</code>	(Short: <code>\p{Jg=Kaf}</code>) (5)
	<code>\p{Joining_Group: Kaph}</code>	(Short: <code>\p{Jg=Kaph}</code>) (1)
	<code>\p{Joining_Group: Khaph}</code>	(Short: <code>\p{Jg=Khaph}</code>) (1)
	<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>) (7)
	<code>\p{Joining_Group: Lamadh}</code>	(Short: <code>\p{Jg=Lamadh}</code>) (1)

```

\p{Joining_Group: Meem} (Short: \p{Jg=Meem}) (4)
\p{Joining_Group: Mim} (Short: \p{Jg=Mim}) (1)
\p{Joining_Group: No_Joining_Group} (Short: \p{Jg=NoJoiningGroup})
(1_113_870)
\p{Joining_Group: Noon} (Short: \p{Jg=Noon}) (8)
\p{Joining_Group: Nun} (Short: \p{Jg=Nun}) (1)
\p{Joining_Group: Nya} (Short: \p{Jg=Nya}) (1)
\p{Joining_Group: Pe} (Short: \p{Jg=Pe}) (1)
\p{Joining_Group: Qaf} (Short: \p{Jg=Qaf}) (5)
\p{Joining_Group: Qaph} (Short: \p{Jg=Qaph}) (1)
\p{Joining_Group: Reh} (Short: \p{Jg=Reh}) (17)
\p{Joining_Group: Reversed_Pe} (Short: \p{Jg=ReversedPe}) (1)
\p{Joining_Group: Rohingya_Yeh} (Short: \p{Jg=RohingyaYeh}) (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}) (4)
\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} \p{Joining_Group=
Hamza_On_Heh_Goal} (1)
\p{Joining_Group: Teth} (Short: \p{Jg=Teth}) (2)
\p{Joining_Group: Waw} (Short: \p{Jg=Waw}) (16)
\p{Joining_Group: Yeh} (Short: \p{Jg=Yeh}) (10)
\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} (215)
\p{Joining_Type: Dual_Joining} (Short: \p{Jt=D}) (215)
\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_389)
\p{Joining_Type: R} \p{Joining_Type=Right_Joining} (82)
\p{Joining_Type: Right_Joining} (Short: \p{Jt=R}) (82)
\p{Joining_Type: T} \p{Joining_Type=Transparent} (1423)
\p{Joining_Type: Transparent} (Short: \p{Jt=T}) (1423)
\p{Joining_Type: U} \p{Joining_Type=Non_Joining} (1_112_389)
\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_Sup} \p{Kana_Supplement} (= \p{Block=
Kana_Supplement}) (256)
X \p{Kana_Supplement} \p{Block=Kana_Supplement} (Short:
\p{InKanaSup}) (256)

```

X	<code>\p{Kanbun}</code>	<code>\p{Block=Kanbun}</code> (16)
X	<code>\p{Kangxi}</code>	<code>\p{Kangxi_Radicals}</code> (= <code>\p{Block=Kangxi_Radicals}</code>) (224)
X	<code>\p{Kangxi_Radicals}</code>	<code>\p{Block=Kangxi_Radicals}</code> (Short: <code>\p{InKangxi}</code>) (224)
	<code>\p{Kannada}</code>	<code>\p{Script=Kannada}</code> (Short: <code>\p{Knda}</code> ; NOT <code>\p{Block=Kannada}</code>) (86)
	<code>\p{Katakana}</code>	<code>\p{Script=Katakana}</code> (Short: <code>\p{Kana}</code> ; NOT <code>\p{Block=Katakana}</code>) (300)
X	<code>\p{Katakana_Ext}</code>	<code>\p{Katakana_Phonetic_Extensions}</code> (= <code>\p{Block=Katakana_Phonetic_Extensions}</code>) (16)
X	<code>\p{Katakana_Phonetic_Extensions}</code>	<code>\p{Block=Katakana_Phonetic_Extensions}</code> (Short: <code>\p{InKatakanaExt}</code>) (16)
	<code>\p{Kayah_Li}</code>	<code>\p{Script=Kayah_Li}</code> (Short: <code>\p{Kali}</code>) (48)
	<code>\p{Khar}</code>	<code>\p{Kharoshthi}</code> (= <code>\p{Script=Kharoshthi}</code>) (NOT <code>\p{Block=Kharoshthi}</code>) (65)
	<code>\p{Kharoshthi}</code>	<code>\p{Script=Kharoshthi}</code> (Short: <code>\p{Khar}</code> ; NOT <code>\p{Block=Kharoshthi}</code>) (65)
	<code>\p{Khmer}</code>	<code>\p{Script=Khmer}</code> (Short: <code>\p{Khmr}</code> ; NOT <code>\p{Block=Khmer}</code>) (146)
X	<code>\p{Khmer_Symbols}</code>	<code>\p{Block=Khmer_Symbols}</code> (32)
	<code>\p{Khmr}</code>	<code>\p{Khmer}</code> (= <code>\p{Script=Khmer}</code>) (NOT <code>\p{Block=Khmer}</code>) (146)
	<code>\p{Knda}</code>	<code>\p{Kannada}</code> (= <code>\p{Script=Kannada}</code>) (NOT <code>\p{Block=Kannada}</code>) (86)
	<code>\p{Kthi}</code>	<code>\p{Kaithi}</code> (= <code>\p{Script=Kaithi}</code>) (NOT <code>\p{Block=Kaithi}</code>) (66)
	<code>\p{L}</code>	<code>\p{Letter}</code> (= <code>\p{General_Category=Letter}</code>) (101_013)
X	<code>\p{L&}</code>	<code>\p{Cased_Letter}</code> (= <code>\p{General_Category=Cased_Letter}</code>) (3223)
X	<code>\p{L_}</code>	<code>\p{Cased_Letter}</code> (= <code>\p{General_Category=Cased_Letter}</code>) Note the trailing '_' matters in spite of loose matching rules. (3223)
	<code>\p{Lana}</code>	<code>\p{Tai_Tham}</code> (= <code>\p{Script=Tai_Tham}</code>) (NOT <code>\p{Block=Tai_Tham}</code>) (127)
	<code>\p{Lao}</code>	<code>\p{Script=Lao}</code> (NOT <code>\p{Block=Lao}</code>) (67)
	<code>\p{Laoo}</code>	<code>\p{Lao}</code> (= <code>\p{Script=Lao}</code>) (NOT <code>\p{Block=Lao}</code>) (67)
	<code>\p{Latin}</code>	<code>\p{Script=Latin}</code> (Short: <code>\p{Latn}</code>) (1272)
X	<code>\p{Latin_1}</code>	<code>\p{Latin_1_Supplement}</code> (= <code>\p{Block=Latin_1_Supplement}</code>) (128)
X	<code>\p{Latin_1_Sup}</code>	<code>\p{Latin_1_Supplement}</code> (= <code>\p{Block=Latin_1_Supplement}</code>) (128)
X	<code>\p{Latin_1_Supplement}</code>	<code>\p{Block=Latin_1_Supplement}</code> (Short: <code>\p{InLatin1}</code>) (128)
X	<code>\p{Latin_Ext_A}</code>	<code>\p{Latin_Extended_A}</code> (= <code>\p{Block=Latin_Extended_A}</code>) (128)
X	<code>\p{Latin_Ext_Additional}</code>	<code>\p{Latin_Extended_Additional}</code> (= <code>\p{Block=Latin_Extended_Additional}</code>) (256)
X	<code>\p{Latin_Ext_B}</code>	<code>\p{Latin_Extended_B}</code> (= <code>\p{Block=Latin_Extended_B}</code>) (208)

X	<code>\p{Latin_Ext_C}</code>	<code>\p{Latin_Extended_C}</code> (= <code>\p{Block=Latin_Extended_C}</code>) (32)
X	<code>\p{Latin_Ext_D}</code>	<code>\p{Latin_Extended_D}</code> (= <code>\p{Block=Latin_Extended_D}</code>) (224)
X	<code>\p{Latin_Extended_A}</code>	<code>\p{Block=Latin_Extended_A}</code> (Short: <code>\p{InLatinExtA}</code>) (128)
X	<code>\p{Latin_Extended_Additional}</code>	<code>\p{Block=Latin_Extended_Additional}</code> (Short: <code>\p{InLatinExtAdditional}</code>) (256)
X	<code>\p{Latin_Extended_B}</code>	<code>\p{Block=Latin_Extended_B}</code> (Short: <code>\p{InLatinExtB}</code>) (208)
X	<code>\p{Latin_Extended_C}</code>	<code>\p{Block=Latin_Extended_C}</code> (Short: <code>\p{InLatinExtC}</code>) (32)
X	<code>\p{Latin_Extended_D}</code>	<code>\p{Block=Latin_Extended_D}</code> (Short: <code>\p{InLatinExtD}</code>) (224)
	<code>\p{Latn}</code>	<code>\p{Latin}</code> (= <code>\p{Script=Latin}</code>) (1272)
	<code>\p{Lb: *}</code>	<code>\p{Line_Break: *}</code>
	<code>\p{LC}</code>	<code>\p{Cased_Letter}</code> (= <code>\p{General_Category=Cased_Letter}</code>) (3223)
	<code>\p{Lepc}</code>	<code>\p{Lepcha}</code> (= <code>\p{Script=Lepcha}</code>) (NOT <code>\p{Block=Lepcha}</code>) (74)
	<code>\p{Lepcha}</code>	<code>\p{Script=Lepcha}</code> (Short: <code>\p{Lepc}</code> ; NOT <code>\p{Block=Lepcha}</code>) (74)
	<code>\p{Letter}</code>	<code>\p{General_Category=Letter}</code> (Short: <code>\p{L}</code>) (101_013)
	<code>\p{Letter_Number}</code>	<code>\p{General_Category=Letter_Number}</code> (Short: <code>\p{NL}</code>) (224)
X	<code>\p{Letterlike_Symbols}</code>	<code>\p{Block=Letterlike_Symbols}</code> (80)
	<code>\p{Limbu}</code>	<code>\p{Limbu}</code> (= <code>\p{Script=Limbu}</code>) (NOT <code>\p{Block=Limbu}</code>) (66)
	<code>\p{Limbu}</code>	<code>\p{Script=Limbu}</code> (Short: <code>\p{Limbu}</code> ; NOT <code>\p{Block=Limbu}</code>) (66)
	<code>\p{Linb}</code>	<code>\p{Linear_B}</code> (= <code>\p{Script=Linear_B}</code>) (211)
	<code>\p{Line_Break: AI}</code>	<code>\p{Line_Break=Ambiguous}</code> (687)
	<code>\p{Line_Break: AL}</code>	<code>\p{Line_Break=Alphabetic}</code> (15_355)
	<code>\p{Line_Break: Alphabetic}</code>	(Short: <code>\p{Lb=AL}</code>) (15_355)
	<code>\p{Line_Break: Ambiguous}</code>	(Short: <code>\p{Lb=AI}</code>) (687)
	<code>\p{Line_Break: B2}</code>	<code>\p{Line_Break=Break_Both}</code> (3)
	<code>\p{Line_Break: BA}</code>	<code>\p{Line_Break=Break_After}</code> (151)
	<code>\p{Line_Break: BB}</code>	<code>\p{Line_Break=Break_Before}</code> (19)
	<code>\p{Line_Break: BK}</code>	<code>\p{Line_Break=Mandatory_Break}</code> (4)
	<code>\p{Line_Break: Break_After}</code>	(Short: <code>\p{Lb=BA}</code>) (151)
	<code>\p{Line_Break: Break_Before}</code>	(Short: <code>\p{Lb=BB}</code>) (19)
	<code>\p{Line_Break: Break_Both}</code>	(Short: <code>\p{Lb=B2}</code>) (3)
	<code>\p{Line_Break: Break_Symbols}</code>	(Short: <code>\p{Lb=SY}</code>) (1)
	<code>\p{Line_Break: Carriage_Return}</code>	(Short: <code>\p{Lb=CR}</code>) (1)
	<code>\p{Line_Break: CB}</code>	<code>\p{Line_Break=Contingent_Break}</code> (1)
	<code>\p{Line_Break: CJ}</code>	<code>\p{Line_Break=Conditional_Japanese_Starter}</code> (51)
	<code>\p{Line_Break: CL}</code>	<code>\p{Line_Break=Close_Punctuation}</code> (87)
	<code>\p{Line_Break: Close_Parenthesis}</code>	(Short: <code>\p{Lb=CP}</code>) (2)
	<code>\p{Line_Break: Close_Punctuation}</code>	(Short: <code>\p{Lb=CL}</code>) (87)
	<code>\p{Line_Break: CM}</code>	<code>\p{Line_Break=Combining_Mark}</code> (1628)
	<code>\p{Line_Break: Combining_Mark}</code>	(Short: <code>\p{Lb=CM}</code>) (1628)
	<code>\p{Line_Break: Complex_Context}</code>	(Short: <code>\p{Lb=SA}</code>) (665)
	<code>\p{Line_Break: Conditional_Japanese_Starter}</code>	(Short: <code>\p{Lb=CJ}</code>) (51)

<code>\p{Line_Break: Contingent_Break}</code>	(Short: <code>\p{Lb=CB}</code>) (1)
<code>\p{Line_Break: CP}</code>	<code>\p{Line_Break=Close_Parenthesis}</code> (2)
<code>\p{Line_Break: CR}</code>	<code>\p{Line_Break=Carriage_Return}</code> (1)
<code>\p{Line_Break: EX}</code>	<code>\p{Line_Break=Exclamation}</code> (34)
<code>\p{Line_Break: Exclamation}</code>	(Short: <code>\p{Lb=EX}</code>) (34)
<code>\p{Line_Break: GL}</code>	<code>\p{Line_Break=Glue}</code> (18)
<code>\p{Line_Break: Glue}</code>	(Short: <code>\p{Lb=GL}</code>) (18)
<code>\p{Line_Break: H2}</code>	(Short: <code>\p{Lb=H2}</code>) (399)
<code>\p{Line_Break: H3}</code>	(Short: <code>\p{Lb=H3}</code>) (10_773)
<code>\p{Line_Break: Hebrew_Letter}</code>	(Short: <code>\p{Lb=HL}</code>) (74)
<code>\p{Line_Break: HL}</code>	<code>\p{Line_Break=Hebrew_Letter}</code> (74)
<code>\p{Line_Break: HY}</code>	<code>\p{Line_Break=Hyphen}</code> (1)
<code>\p{Line_Break: Hyphen}</code>	(Short: <code>\p{Lb=HY}</code>) (1)
<code>\p{Line_Break: ID}</code>	<code>\p{Line_Break=Ideographic}</code> (162_700)
<code>\p{Line_Break: Ideographic}</code>	(Short: <code>\p{Lb=ID}</code>) (162_700)
<code>\p{Line_Break: IN}</code>	<code>\p{Line_Break=Inseparable}</code> (4)
<code>\p{Line_Break: Infix_Numeric}</code>	(Short: <code>\p{Lb=IS}</code>) (13)
<code>\p{Line_Break: Inseparable}</code>	(Short: <code>\p{Lb=IN}</code>) (4)
<code>\p{Line_Break: Inseperable}</code>	<code>\p{Line_Break=Inseparable}</code> (4)
<code>\p{Line_Break: IS}</code>	<code>\p{Line_Break=Infix_Numeric}</code> (13)
<code>\p{Line_Break: JL}</code>	(Short: <code>\p{Lb=JL}</code>) (125)
<code>\p{Line_Break: JT}</code>	(Short: <code>\p{Lb=JT}</code>) (137)
<code>\p{Line_Break: JV}</code>	(Short: <code>\p{Lb=JV}</code>) (95)
<code>\p{Line_Break: LF}</code>	<code>\p{Line_Break=Line_Feed}</code> (1)
<code>\p{Line_Break: Line_Feed}</code>	(Short: <code>\p{Lb=LF}</code>) (1)
<code>\p{Line_Break: Mandatory_Break}</code>	(Short: <code>\p{Lb=BK}</code>) (4)
<code>\p{Line_Break: Next_Line}</code>	(Short: <code>\p{Lb=NL}</code>) (1)
<code>\p{Line_Break: NL}</code>	<code>\p{Line_Break=Next_Line}</code> (1)
<code>\p{Line_Break: Nonstarter}</code>	(Short: <code>\p{Lb=NS}</code>) (26)
<code>\p{Line_Break: NS}</code>	<code>\p{Line_Break=Nonstarter}</code> (26)
<code>\p{Line_Break: NU}</code>	<code>\p{Line_Break=Numeric}</code> (452)
<code>\p{Line_Break: Numeric}</code>	(Short: <code>\p{Lb=NU}</code>) (452)
<code>\p{Line_Break: OP}</code>	<code>\p{Line_Break=Open_Punctuation}</code> (81)
<code>\p{Line_Break: Open_Punctuation}</code>	(Short: <code>\p{Lb=OP}</code>) (81)
<code>\p{Line_Break: PO}</code>	<code>\p{Line_Break=Postfix_Numeric}</code> (28)
<code>\p{Line_Break: Postfix_Numeric}</code>	(Short: <code>\p{Lb=PO}</code>) (28)
<code>\p{Line_Break: PR}</code>	<code>\p{Line_Break=Prefix_Numeric}</code> (46)
<code>\p{Line_Break: Prefix_Numeric}</code>	(Short: <code>\p{Lb=PR}</code>) (46)
<code>\p{Line_Break: QU}</code>	<code>\p{Line_Break=Quotation}</code> (34)
<code>\p{Line_Break: Quotation}</code>	(Short: <code>\p{Lb=QU}</code>) (34)
<code>\p{Line_Break: Regional_Indicator}</code>	(Short: <code>\p{Lb=RI}</code>) (26)
<code>\p{Line_Break: RI}</code>	<code>\p{Line_Break=Regional_Indicator}</code> (26)
<code>\p{Line_Break: SA}</code>	<code>\p{Line_Break=Complex_Context}</code> (665)
D <code>\p{Line_Break: SG}</code>	<code>\p{Line_Break=Surrogate}</code> (2048)
<code>\p{Line_Break: SP}</code>	<code>\p{Line_Break=Space}</code> (1)
<code>\p{Line_Break: Space}</code>	(Short: <code>\p{Lb=SP}</code>) (1)
D <code>\p{Line_Break: Surrogate}</code>	Deprecated by Unicode because surrogates should never appear in well-formed text, and therefore shouldn't be the basis for line breaking (Short: <code>\p{Lb=SG}</code>) (2048)
<code>\p{Line_Break: SY}</code>	<code>\p{Line_Break=Break_Symbols}</code> (1)
<code>\p{Line_Break: Unknown}</code>	(Short: <code>\p{Lb=XX}</code>) (918_337)
<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> (918_337)
<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>) (1751)
	<code>\p{Lm}</code>	<code>\p{Modifier_Letter}</code> (= <code>\p{General_Category=Modifier_Letter}</code>) (237)
	<code>\p{Lo}</code>	<code>\p{Other_Letter}</code> (= <code>\p{General_Category=Other_Letter}</code>) (97_553)
	<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>) (1934)
	<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>) (1934)
	<code>\p{Lowercase: N*}</code>	(Short: <code>\p{Lower=N}</code> , <code>\P{Lower}</code> ; /i= <code>Cased=No</code>) (1_112_178)
	<code>\p{Lowercase: Y*}</code>	(Short: <code>\p{Lower=Y}</code> , <code>\P{Lower}</code> ; /i= <code>Cased=Yes</code>) (1934)
	<code>\p{Lowercase_Letter}</code>	<code>\p{General_Category=Lowercase_Letter}</code> (Short: <code>\p{Ll}</code> ; /i= <code>General_Category=Cased_Letter</code>) (1751)
	<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>) (1441)
	<code>\p{Lyci}</code>	<code>\p{Lycian}</code> (= <code>\p{Script=Lycian}</code>) (NOT <code>\p{Block=Lycian}</code>) (29)
	<code>\p{Lycian}</code>	<code>\p{Script=Lycian}</code> (Short: <code>\p{Lyci}</code> ; NOT <code>\p{Block=Lycian}</code>) (29)
	<code>\p{Lydi}</code>	<code>\p{Lydian}</code> (= <code>\p{Script=Lydian}</code>) (NOT <code>\p{Block=Lydian}</code>) (27)
	<code>\p{Lydian}</code>	<code>\p{Script=Lydian}</code> (Short: <code>\p{Lydi}</code> ; NOT <code>\p{Block=Lydian}</code>) (27)
	<code>\p{M}</code>	<code>\p{Mark}</code> (= <code>\p{General_Category=Mark}</code>) (1645)
X	<code>\p{Mahjong}</code>	<code>\p{Mahjong_Tiles}</code> (= <code>\p{Block=Mahjong_Tiles}</code>) (48)
X	<code>\p{Mahjong_Tiles}</code>	<code>\p{Block=Mahjong_Tiles}</code> (Short:

	<code>\p{InMahjong}</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>) (1645)
<code>\p{Math}</code>	<code>\p{Math=Y}</code> (2310)
<code>\p{Math: N*}</code>	(Single: <code>\p{Math}</code>) (1_111_802)
<code>\p{Math: Y*}</code>	(Single: <code>\p{Math}</code>) (2310)
X <code>\p{Math_Alphanum}</code>	<code>\p{Mathematical_Alphanumeric_Symbols}</code> (= <code>\p{Block=Mathematical_Alphanumeric_Symbols}</code>) (1024)
X <code>\p{Math_Operators}</code>	<code>\p{Mathematical_Operators}</code> (= <code>\p{Block=Mathematical_Operators}</code>) (256)
<code>\p{Math_Symbol}</code>	<code>\p{General_Category=Math_Symbol}</code> (Short: <code>\p{Sm}</code>) (952)
X <code>\p{Mathematical_Alphanumeric_Symbols}</code>	<code>\p{Block=Mathematical_Alphanumeric_Symbols}</code> (Short: <code>\p{InMathAlphanum}</code>) (1024)
X <code>\p{Mathematical_Operators}</code>	<code>\p{Block=Mathematical_Operators}</code> (Short: <code>\p{InMathOperators}</code>) (256)
<code>\p{Mc}</code>	<code>\p{Spacing_Mark}</code> (= <code>\p{General_Category=Spacing_Mark}</code>) (353)
<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>) (79)
X <code>\p{Meetei_Mayek_Ext}</code>	<code>\p{Meetei_Mayek_Extensions}</code> (= <code>\p{Block=Meetei_Mayek_Extensions}</code>) (32)
X <code>\p{Meetei_Mayek_Extensions}</code>	<code>\p{Block=Meetei_Mayek_Extensions}</code> (Short: <code>\p{InMeeteiMayekExt}</code>) (32)
<code>\p{Merc}</code>	<code>\p{Meroitic_Cursive}</code> (= <code>\p{Script=Meroitic_Cursive}</code>) (NOT <code>\p{Block=Meroitic_Cursive}</code>) (26)
<code>\p{Mero}</code>	<code>\p{Meroitic_Hieroglyphs}</code> (= <code>\p{Script=Meroitic_Hieroglyphs}</code>) (32)
<code>\p{Meroitic_Cursive}</code>	<code>\p{Script=Meroitic_Cursive}</code> (Short: <code>\p{Merc}</code> ; NOT <code>\p{Block=Meroitic_Cursive}</code>) (26)
<code>\p{Meroitic_Hieroglyphs}</code>	<code>\p{Script=Meroitic_Hieroglyphs}</code> (Short: <code>\p{Mero}</code>) (32)
<code>\p{Miao}</code>	<code>\p{Script=Miao}</code> (NOT <code>\p{Block=Miao}</code>) (133)
X <code>\p{Misc_Arrows}</code>	<code>\p{Miscellaneous_Symbols_And_Arrows}</code> (= <code>\p{Block=Miscellaneous_Symbols_And_Arrows}</code>) (256)
X <code>\p{Misc_Math_Symbols_A}</code>	<code>\p{Miscellaneous_Mathematical_Symbols_A}</code> (= <code>\p{Block=Miscellaneous_Mathematical_Symbols_A}</code>) (48)
X <code>\p{Misc_Math_Symbols_B}</code>	<code>\p{Miscellaneous_Mathematical_Symbols_B}</code> (= <code>\p{Block=Miscellaneous_Mathematical_Symbols_B}</code>)

	(128)
X <code>\p{Misc_Pictographs}</code>	<code>\p{Miscellaneous_Symbols_And_Pictographs}</code> (= <code>\p{Block=Miscellaneous_Symbols_And_Pictographs}</code>) (768)
X <code>\p{Misc_Symbols}</code>	<code>\p{Miscellaneous_Symbols}</code> (= <code>\p{Block=Miscellaneous_Symbols}</code>) (256)
X <code>\p{Misc_Technical}</code>	<code>\p{Miscellaneous_Technical}</code> (= <code>\p{Block=Miscellaneous_Technical}</code>) (256)
X <code>\p{Miscellaneous_Mathematical_Symbols_A}</code>	<code>\p{Block=Miscellaneous_Mathematical_Symbols_A}</code> (Short: <code>\p{InMiscMathSymbolsA}</code>) (48)
X <code>\p{Miscellaneous_Mathematical_Symbols_B}</code>	<code>\p{Block=Miscellaneous_Mathematical_Symbols_B}</code> (Short: <code>\p{InMiscMathSymbolsB}</code>) (128)
X <code>\p{Miscellaneous_Symbols}</code>	<code>\p{Block=Miscellaneous_Symbols}</code> (Short: <code>\p{InMiscSymbols}</code>) (256)
X <code>\p{Miscellaneous_Symbols_And_Arrows}</code>	<code>\p{Block=Miscellaneous_Symbols_And_Arrows}</code> (Short: <code>\p{InMiscArrows}</code>) (256)
X <code>\p{Miscellaneous_Symbols_And_Pictographs}</code>	<code>\p{Block=Miscellaneous_Symbols_And_Pictographs}</code> (Short: <code>\p{InMiscPictographs}</code>) (768)
X <code>\p{Miscellaneous_Technical}</code>	<code>\p{Block=Miscellaneous_Technical}</code> (Short: <code>\p{InMiscTechnical}</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>) (1280)
<code>\p{Modifier_Letter}</code>	<code>\p{General_Category=Modifier_Letter}</code> (Short: <code>\p{Lm}</code>) (237)
X <code>\p{Modifier_Letters}</code>	<code>\p{Spacing_Modifier_Letters}</code> (= <code>\p{Block=Spacing_Modifier_Letters}</code>) (80)
<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 <code>\p{Block=Mongolian}</code>) (153)
<code>\p{Mongolian}</code>	<code>\p{Script=Mongolian}</code> (Short: <code>\p{Mong}</code> ; NOT <code>\p{Block=Mongolian}</code>) (153)
<code>\p{Mtei}</code>	<code>\p{Meetei_Mayek}</code> (= <code>\p{Script=Meetei_Mayek}</code>) (NOT <code>\p{Block=Meetei_Mayek}</code>) (79)
X <code>\p{Music}</code>	<code>\p{Musical_Symbols}</code> (= <code>\p{Block=Musical_Symbols}</code>) (256)
X <code>\p{Musical_Symbols}</code>	<code>\p{Block=Musical_Symbols}</code> (Short: <code>\p{InMusic}</code>) (256)
<code>\p{Myanmar}</code>	<code>\p{Script=Myanmar}</code> (Short: <code>\p{Mymr}</code> ; NOT <code>\p{Block=Myanmar}</code>) (188)
X <code>\p{Myanmar_Ext_A}</code>	<code>\p{Myanmar_Extended_A}</code> (= <code>\p{Block=Myanmar_Extended_A}</code>) (32)
X <code>\p{Myanmar_Extended_A}</code>	<code>\p{Block=Myanmar_Extended_A}</code> (Short: <code>\p{InMyanmarExtA}</code>) (32)
<code>\p{Mymr}</code>	<code>\p{Myanmar}</code> (= <code>\p{Script=Myanmar}</code>) (NOT <code>\p{Block=Myanmar}</code>) (188)

<code>\p{N}</code>	<code>\p{Number}</code> (= <code>\p{General_Category=Number}</code>) (1148)
X <code>\p{NB}</code>	<code>\p{No_Block}</code> (= <code>\p{Block=No_Block}</code>) (860_672)
<code>\p{NChar}</code>	<code>\p{Noncharacter_Code_Point}</code> (= <code>\p{Noncharacter_Code_Point=Y}</code>) (66)
<code>\p{NChar: *}</code>	<code>\p{Noncharacter_Code_Point: *}</code>
<code>\p{Nd}</code>	<code>\p{Digit}</code> (= <code>\p{General_Category=Decimal_Number}</code>) (460)
<code>\p{New_Tai_Lue}</code>	<code>\p{Script=New_Tai_Lue}</code> (Short: <code>\p{Talu}</code> ; NOT <code>\p{Block=New_Tai_Lue}</code>) (83)
<code>\p{NFC_QC: *}</code>	<code>\p{NFC_Quick_Check: *}</code>
<code>\p{NFC_Quick_Check: M}</code>	<code>\p{NFC_Quick_Check=Maybe}</code> (104)
<code>\p{NFC_Quick_Check: Maybe}</code>	(Short: <code>\p{NFCQC=M}</code>) (104)
<code>\p{NFC_Quick_Check: N}</code>	<code>\p{NFC_Quick_Check=No}</code> (NOT <code>\P{NFC_Quick_Check}</code> NOR <code>\P{NFC_QC}</code>) (1120)
<code>\p{NFC_Quick_Check: No}</code>	(Short: <code>\p{NFCQC=N}</code> ; NOT <code>\P{NFC_Quick_Check}</code> NOR <code>\P{NFC_QC}</code>) (1120)
<code>\p{NFC_Quick_Check: Y}</code>	<code>\p{NFC_Quick_Check=Yes}</code> (NOT <code>\p{NFC_Quick_Check}</code> NOR <code>\p{NFC_QC}</code>) (1_112_888)
<code>\p{NFC_Quick_Check: Yes}</code>	(Short: <code>\p{NFCQC=Y}</code> ; NOT <code>\p{NFC_Quick_Check}</code> NOR <code>\p{NFC_QC}</code>) (1_112_888)
<code>\p{NFD_QC: *}</code>	<code>\p{NFD_Quick_Check: *}</code>
<code>\p{NFD_Quick_Check: N}</code>	<code>\p{NFD_Quick_Check=No}</code> (NOT <code>\P{NFD_Quick_Check}</code> NOR <code>\P{NFD_QC}</code>) (13_225)
<code>\p{NFD_Quick_Check: No}</code>	(Short: <code>\p{NFDQC=N}</code> ; NOT <code>\P{NFD_Quick_Check}</code> NOR <code>\P{NFD_QC}</code>) (13_225)
<code>\p{NFD_Quick_Check: Y}</code>	<code>\p{NFD_Quick_Check=Yes}</code> (NOT <code>\p{NFD_Quick_Check}</code> NOR <code>\p{NFD_QC}</code>) (1_100_887)
<code>\p{NFD_Quick_Check: Yes}</code>	(Short: <code>\p{NFDQC=Y}</code> ; NOT <code>\p{NFD_Quick_Check}</code> NOR <code>\p{NFD_QC}</code>) (1_100_887)
<code>\p{NFKC_QC: *}</code>	<code>\p{NFKC_Quick_Check: *}</code>
<code>\p{NFKC_Quick_Check: M}</code>	<code>\p{NFKC_Quick_Check=Maybe}</code> (104)
<code>\p{NFKC_Quick_Check: Maybe}</code>	(Short: <code>\p{NFKCQC=M}</code>) (104)
<code>\p{NFKC_Quick_Check: N}</code>	<code>\p{NFKC_Quick_Check=No}</code> (NOT <code>\P{NFKC_Quick_Check}</code> NOR <code>\P{NFKC_QC}</code>) (4787)
<code>\p{NFKC_Quick_Check: No}</code>	(Short: <code>\p{NFKCQC=N}</code> ; NOT <code>\P{NFKC_Quick_Check}</code> NOR <code>\P{NFKC_QC}</code>) (4787)
<code>\p{NFKC_Quick_Check: Y}</code>	<code>\p{NFKC_Quick_Check=Yes}</code> (NOT <code>\p{NFKC_Quick_Check}</code> NOR <code>\p{NFKC_QC}</code>) (1_109_221)
<code>\p{NFKC_Quick_Check: Yes}</code>	(Short: <code>\p{NFKCQC=Y}</code> ; NOT <code>\p{NFKC_Quick_Check}</code> NOR <code>\p{NFKC_QC}</code>) (1_109_221)
<code>\p{NFKD_QC: *}</code>	<code>\p{NFKD_Quick_Check: *}</code>
<code>\p{NFKD_Quick_Check: N}</code>	<code>\p{NFKD_Quick_Check=No}</code> (NOT

	<code>\P{NFKD_Quick_Check}</code>	<code>NOR \P{NFKD_QC}</code>	
	<code>(16_880)</code>		
	<code>\p{NFKD_Quick_Check: No}</code>	<code>(Short: \p{NFKDQC=N}; NOT</code>	
	<code>\P{NFKD_Quick_Check}</code>	<code>NOR \P{NFKD_QC}</code>	
	<code>(16_880)</code>		
	<code>\p{NFKD_Quick_Check: Y}</code>	<code>\p{NFKD_Quick_Check=Yes}</code>	<code>(NOT</code>
	<code>\p{NFKD_Quick_Check}</code>	<code>NOR \P{NFKD_QC}</code>	
	<code>(1_097_232)</code>		
	<code>\p{NFKD_Quick_Check: Yes}</code>	<code>(Short: \p{NFKDQC=Y}; NOT</code>	
	<code>\p{NFKD_Quick_Check}</code>	<code>NOR \P{NFKD_QC}</code>	
	<code>(1_097_232)</code>		
	<code>\p{Nko}</code>	<code>\p{Script=Nko}</code>	<code>(NOT \p{Nko}) (59)</code>
	<code>\p{Nkoo}</code>	<code>\p{Nko}</code>	<code>(= \p{Script=Nko}) (NOT \p{Nko})</code>
		<code>(59)</code>	
	<code>\p{Nl}</code>	<code>\p{Letter_Number}</code>	<code>(= \p{General_Category=</code>
		<code>Letter_Number}) (224)</code>	
	<code>\p{No}</code>	<code>\p{Other_Number}</code>	<code>(= \p{General_Category=</code>
		<code>Other_Number}) (464)</code>	
X	<code>\p{No_Block}</code>	<code>\p{Block=No_Block}</code>	<code>(Short: \p{InNB})</code>
		<code>(860_672)</code>	
	<code>\p{Noncharacter_Code_Point}</code>	<code>\p{Noncharacter_Code_Point=Y}</code>	<code>(Short:</code>
		<code>\p{NChar}) (66)</code>	
	<code>\p{Noncharacter_Code_Point: N*}</code>	<code>(Short: \p{NChar=N}, \P{NChar})</code>	
		<code>(1_114_046)</code>	
	<code>\p{Noncharacter_Code_Point: Y*}</code>	<code>(Short: \p{NChar=Y}, \P{NChar})</code>	
		<code>(66)</code>	
	<code>\p{Nonspacing_Mark}</code>	<code>\p{General_Category=Nonspacing_Mark}</code>	
		<code>(Short: \p{Mn}) (1280)</code>	
	<code>\p{Nt: *}</code>	<code>\p{Numeric_Type: *}</code>	
	<code>\p{Number}</code>	<code>\p{General_Category=Number}</code>	<code>(Short: \p{N})</code>
		<code>(1148)</code>	
X	<code>\p{Number_Forms}</code>	<code>\p{Block=Number_Forms}</code>	<code>(64)</code>
	<code>\p{Numeric_Type: De}</code>	<code>\p{Numeric_Type=Decimal}</code>	<code>(460)</code>
	<code>\p{Numeric_Type: Decimal}</code>	<code>(Short: \p{Nt=De}) (460)</code>	
	<code>\p{Numeric_Type: Di}</code>	<code>\p{Numeric_Type=Digit}</code>	<code>(128)</code>
	<code>\p{Numeric_Type: Digit}</code>	<code>(Short: \p{Nt=Di}) (128)</code>	
	<code>\p{Numeric_Type: None}</code>	<code>(Short: \p{Nt=None}) (1_112_883)</code>	
	<code>\p{Numeric_Type: Nu}</code>	<code>\p{Numeric_Type=Numeric}</code>	<code>(641)</code>
	<code>\p{Numeric_Type: Numeric}</code>	<code>(Short: \p{Nt=Nu}) (641)</code>	
T	<code>\p{Numeric_Value: -1}</code>	<code>(Short: \p{Nv=-1}) (2)</code>	
T	<code>\p{Numeric_Value: -1/2}</code>	<code>(Short: \p{Nv=-1/2}) (1)</code>	
T	<code>\p{Numeric_Value: 0}</code>	<code>(Short: \p{Nv=0}) (60)</code>	
T	<code>\p{Numeric_Value: 1/16}</code>	<code>(Short: \p{Nv=1/16}) (3)</code>	
T	<code>\p{Numeric_Value: 1/10}</code>	<code>(Short: \p{Nv=1/10}) (1)</code>	
T	<code>\p{Numeric_Value: 1/9}</code>	<code>(Short: \p{Nv=1/9}) (1)</code>	
T	<code>\p{Numeric_Value: 1/8}</code>	<code>(Short: \p{Nv=1/8}) (5)</code>	
T	<code>\p{Numeric_Value: 1/7}</code>	<code>(Short: \p{Nv=1/7}) (1)</code>	
T	<code>\p{Numeric_Value: 1/6}</code>	<code>(Short: \p{Nv=1/6}) (2)</code>	
T	<code>\p{Numeric_Value: 3/16}</code>	<code>(Short: \p{Nv=3/16}) (3)</code>	
T	<code>\p{Numeric_Value: 1/5}</code>	<code>(Short: \p{Nv=1/5}) (1)</code>	
T	<code>\p{Numeric_Value: 1/4}</code>	<code>(Short: \p{Nv=1/4}) (9)</code>	
T	<code>\p{Numeric_Value: 1/3}</code>	<code>(Short: \p{Nv=1/3}) (4)</code>	
T	<code>\p{Numeric_Value: 3/8}</code>	<code>(Short: \p{Nv=3/8}) (1)</code>	
T	<code>\p{Numeric_Value: 2/5}</code>	<code>(Short: \p{Nv=2/5}) (1)</code>	
T	<code>\p{Numeric_Value: 1/2}</code>	<code>(Short: \p{Nv=1/2}) (10)</code>	
T	<code>\p{Numeric_Value: 3/5}</code>	<code>(Short: \p{Nv=3/5}) (1)</code>	

T	<code>\p{Numeric_Value: 43}</code>	(Short: <code>\p{Nv=43}</code>) (1)
T	<code>\p{Numeric_Value: 44}</code>	(Short: <code>\p{Nv=44}</code>) (1)
T	<code>\p{Numeric_Value: 45}</code>	(Short: <code>\p{Nv=45}</code>) (1)
T	<code>\p{Numeric_Value: 46}</code>	(Short: <code>\p{Nv=46}</code>) (1)
T	<code>\p{Numeric_Value: 47}</code>	(Short: <code>\p{Nv=47}</code>) (1)
T	<code>\p{Numeric_Value: 48}</code>	(Short: <code>\p{Nv=48}</code>) (1)
T	<code>\p{Numeric_Value: 49}</code>	(Short: <code>\p{Nv=49}</code>) (1)
T	<code>\p{Numeric_Value: 50}</code>	(Short: <code>\p{Nv=50}</code>) (20)
T	<code>\p{Numeric_Value: 60}</code>	(Short: <code>\p{Nv=60}</code>) (6)
T	<code>\p{Numeric_Value: 70}</code>	(Short: <code>\p{Nv=70}</code>) (6)
T	<code>\p{Numeric_Value: 80}</code>	(Short: <code>\p{Nv=80}</code>) (6)
T	<code>\p{Numeric_Value: 90}</code>	(Short: <code>\p{Nv=90}</code>) (6)
T	<code>\p{Numeric_Value: 100}</code>	(Short: <code>\p{Nv=100}</code>) (20)
T	<code>\p{Numeric_Value: 200}</code>	(Short: <code>\p{Nv=200}</code>) (2)
T	<code>\p{Numeric_Value: 300}</code>	(Short: <code>\p{Nv=300}</code>) (3)
T	<code>\p{Numeric_Value: 400}</code>	(Short: <code>\p{Nv=400}</code>) (2)
T	<code>\p{Numeric_Value: 500}</code>	(Short: <code>\p{Nv=500}</code>) (12)
T	<code>\p{Numeric_Value: 600}</code>	(Short: <code>\p{Nv=600}</code>) (2)
T	<code>\p{Numeric_Value: 700}</code>	(Short: <code>\p{Nv=700}</code>) (2)
T	<code>\p{Numeric_Value: 800}</code>	(Short: <code>\p{Nv=800}</code>) (2)
T	<code>\p{Numeric_Value: 900}</code>	(Short: <code>\p{Nv=900}</code>) (3)
T	<code>\p{Numeric_Value: 1000}</code>	(Short: <code>\p{Nv=1000}</code>) (17)
T	<code>\p{Numeric_Value: 2000}</code>	(Short: <code>\p{Nv=2000}</code>) (1)
T	<code>\p{Numeric_Value: 3000}</code>	(Short: <code>\p{Nv=3000}</code>) (1)
T	<code>\p{Numeric_Value: 4000}</code>	(Short: <code>\p{Nv=4000}</code>) (1)
T	<code>\p{Numeric_Value: 5000}</code>	(Short: <code>\p{Nv=5000}</code>) (5)
T	<code>\p{Numeric_Value: 6000}</code>	(Short: <code>\p{Nv=6000}</code>) (1)
T	<code>\p{Numeric_Value: 7000}</code>	(Short: <code>\p{Nv=7000}</code>) (1)
T	<code>\p{Numeric_Value: 8000}</code>	(Short: <code>\p{Nv=8000}</code>) (1)
T	<code>\p{Numeric_Value: 9000}</code>	(Short: <code>\p{Nv=9000}</code>) (1)
T	<code>\p{Numeric_Value: 10000}</code>	(= 1.0e+04) (Short: <code>\p{Nv=10000}</code>) (7)
T	<code>\p{Numeric_Value: 20000}</code>	(= 2.0e+04) (Short: <code>\p{Nv=20000}</code>) (1)
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: 216000}</code>	(= 2.2e+05) (Short: <code>\p{Nv=216000}</code>) (1)
T	<code>\p{Numeric_Value: 432000}</code>	(= 4.3e+05) (Short: <code>\p{Nv=432000}</code>) (1)
T	<code>\p{Numeric_Value: 100000000}</code>	(= 1.0e+08) (Short: <code>\p{Nv=100000000}</code>) (2)
T	<code>\p{Numeric_Value: 100000000000}</code>	(= 1.0e+12) (Short: <code>\p{Nv=100000000000}</code>) (1)
	<code>\p{Numeric_Value: NaN}</code>	(Short: <code>\p{Nv=NaN}</code>) (1_112_883)
	<code>\p{Nv: *}</code>	<code>\p{Numeric_Value: *}</code>
X	<code>\p{OCR}</code>	<code>\p{Optical_Character_Recognition}</code> (= <code>\p{Block=Optical_Character_Recognition}</code>) (32)
	<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)
	<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{Sarb}</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)
<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> (Short: <code>\p{InOCR}</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 <code>\p{Block=Old_Turkic}</code>) (73)
<code>\p{Orya}</code>	<code>\p{Oriya}</code> (= <code>\p{Script=Oriya}</code>) (NOT <code>\p{Block=Oriya}</code>) (90)
<code>\p{Osma}</code>	<code>\p{Osmanya}</code> (= <code>\p{Script=Osmanya}</code>) (NOT <code>\p{Block=Osmanya}</code>) (40)
<code>\p{Osmanya}</code>	<code>\p{Script=Osmanya}</code> (Short: <code>\p{Osma}</code> ; NOT <code>\p{Block=Osmanya}</code>) (40)
<code>\p{Other}</code>	<code>\p{General_Category=Other}</code> (Short: <code>\p{C}</code>) (1_004_134)
<code>\p{Other_Letter}</code>	<code>\p{General_Category=Other_Letter}</code> (Short: <code>\p{Lo}</code>) (97_553)
<code>\p{Other_Number}</code>	<code>\p{General_Category=Other_Number}</code> (Short: <code>\p{No}</code>) (464)
<code>\p{Other_Punctuation}</code>	<code>\p{General_Category=Other_Punctuation}</code> (Short: <code>\p{Po}</code>) (434)
<code>\p{Other_Symbol}</code>	<code>\p{General_Category=Other_Symbol}</code> (Short: <code>\p{So}</code>) (4404)
<code>\p{P}</code>	<code>\p{Punct}</code> (= <code>\p{General_Category=</code> <code>Punctuation}</code>) (NOT <code>\p{General_Punctuation}</code>) (632)
<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=</code> <code>Y}</code>) (2760)
<code>\p{Pat_Syn: *}</code>	<code>\p{Pattern_Syntax: *}</code>
<code>\p{Pat_WS}</code>	<code>\p{Pattern_White_Space}</code> (= <code>\p{Pattern_White_Space=Y}</code>) (11)
<code>\p{Pat_WS: *}</code>	<code>\p{Pattern_White_Space: *}</code>
<code>\p{Pattern_Syntax}</code>	<code>\p{Pattern_Syntax=Y}</code> (Short: <code>\p{PatSyn}</code>) (2760)
<code>\p{Pattern_Syntax: N*}</code>	(Short: <code>\p{PatSyn=N}</code> , <code>\P{PatSyn}</code>) (1_111_352)
<code>\p{Pattern_Syntax: Y*}</code>	(Short: <code>\p{PatSyn=Y}</code> , <code>\p{PatSyn}</code>) (2760)
<code>\p{Pattern_White_Space}</code>	<code>\p{Pattern_White_Space=Y}</code> (Short: <code>\p{PatWS}</code>) (11)
<code>\p{Pattern_White_Space: N*}</code>	(Short: <code>\p{PatWS=N}</code> , <code>\P{PatWS}</code>) (1_114_101)
<code>\p{Pattern_White_Space: Y*}</code>	(Short: <code>\p{PatWS=Y}</code> , <code>\p{PatWS}</code>) (11)
<code>\p{Pc}</code>	<code>\p{Connector_Punctuation}</code> (=

	<code>\p{General_Category=Connector_Punctuation}</code>) (10)
<code>\p{Pd}</code>	<code>\p{Dash_Punctuation}</code> (= <code>\p{General_Category=Dash_Punctuation}</code>) (23)
<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 = [<code>\f\n\r\t</code>] plus vertical tab (6)
<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}</code>	<code>\p{Phaistos_Disc}</code> (= <code>\p{Block=Phaistos_Disc}</code>) (48)
X <code>\p{Phaistos_Disc}</code>	<code>\p{Block=Phaistos_Disc}</code> (Short: <code>\p{InPhaistos}</code>) (48)
<code>\p{Phli}</code>	<code>\p{Inscriptional_Pahlavi}</code> (= <code>\p{Script=Inscriptional_Pahlavi}</code>) (NOT <code>\p{Block=Inscriptional_Pahlavi}</code>) (27)
<code>\p{Phnx}</code>	<code>\p{Phoenician}</code> (= <code>\p{Script=Phoenician}</code>) (NOT <code>\p{Block=Phoenician}</code>) (29)
<code>\p{Phoenician}</code>	<code>\p{Script=Phoenician}</code> (Short: <code>\p{Phnx}</code> ; NOT <code>\p{Block=Phoenician}</code>) (29)
X <code>\p{Phonetic_Ext}</code>	<code>\p{Phonetic_Extensions}</code> (= <code>\p{Block=Phonetic_Extensions}</code>) (128)
X <code>\p{Phonetic_Ext_Sup}</code>	<code>\p{Phonetic_Extensions_Supplement}</code> (= <code>\p{Block=Phonetic_Extensions_Supplement}</code>) (64)
X <code>\p{Phonetic_Extensions}</code>	<code>\p{Block=Phonetic_Extensions}</code> (Short: <code>\p{InPhoneticExt}</code>) (128)
X <code>\p{Phonetic_Extensions_Supplement}</code>	<code>\p{Block=Phonetic_Extensions_Supplement}</code> (Short: <code>\p{InPhoneticExtSup}</code>) (64)
<code>\p{Pi}</code>	<code>\p{Initial_Punctuation}</code> (= <code>\p{General_Category=Initial_Punctuation}</code>) (12)
X <code>\p{Playing_Cards}</code>	<code>\p{Block=Playing_Cards}</code> (96)
<code>\p{Plrd}</code>	<code>\p{Miao}</code> (= <code>\p{Script=Miao}</code>) (NOT <code>\p{Block=Miao}</code>) (133)
<code>\p{Po}</code>	<code>\p{Other_Punctuation}</code> (= <code>\p{General_Category=Other_Punctuation}</code>) (434)
<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>	<code>[0-9]</code> (10)
<code>\p{PosixGraph}</code>	<code>[-!"#\$%&'()*+,-./:;<>?@[\\]^_`{ }~0-9A-Za-z]</code> (94)
<code>\p{PosixLower}</code>	<code>[a-z]</code> (/i= PosixAlpha) (26)
<code>\p{PosixPrint}</code>	<code>[- 0-9A-Za-z!"#\$%&'()*+,-./:;<>?@[\\]^_`{ }~]</code> (95)
<code>\p{PosixPunct}</code>	<code>[-!"#\$%&'()*+,-./:;<>?@[\\]^_`{ }~]</code> (32)
<code>\p{PosixSpace}</code>	<code>\t, \n, \cK, \f, \r, and ' '</code> . (\cK is vertical tab) (6)
<code>\p{PosixUpper}</code>	<code>[A-Z]</code> (/i= PosixAlpha) (26)
<code>\p{PosixWord}</code>	<code>\p{PerlWord}</code> (63)
<code>\p{PosixXDigit}</code>	<code>\p{ASCII_Hex_Digit=Y}</code> <code>[0-9A-Fa-f]</code> (Short: <code>\p{AHex}</code>) (22)
T <code>\p{Present_In: 1.1}</code>	<code>\p{Age=V1_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 <code>\p{Present_In: 6.0}</code>	Code point's usage introduced in version 6.0 or earlier (Short: <code>\p{In=6.0}</code>) (Perl extension) (249_031)
T <code>\p{Present_In: 6.1}</code>	Code point's usage introduced in version 6.1 or earlier (Short: <code>\p{In=6.1}</code>) (Perl extension) (249_763)
T <code>\p{Present_In: 6.2}</code>	Code point's usage introduced in version 6.2 or earlier (Short: <code>\p{In=6.2}</code>) (Perl extension) (249_764)
<code>\p{Present_In: Unassigned}</code>	<code>\p{Age=Unassigned}</code> (Short: <code>\p{In=Unassigned}</code>) (Perl extension) (864_348)

<code>\p{Print}</code>	Characters that are graphical plus space characters (but no controls) (247_583)
<code>\p{Private_Use}</code>	<code>\p{General_Category=Private_Use}</code> (Short: <code>\p{Co}</code> ; NOT <code>\p{Private_Use_Area}</code>) (137_468)
X <code>\p{Private_Use_Area}</code>	<code>\p{Block=Private_Use_Area}</code> (Short: <code>\p{InPUA}</code>) (6400)
<code>\p{Prti}</code>	<code>\p{Inscriptional_Parthian}</code> (= <code>\p{Script=Inscriptional_Parthian}</code>) (NOT <code>\p{Block=Inscriptional_Parthian}</code>) (30)
<code>\p{Ps}</code>	<code>\p{Open_Punctuation}</code> (= <code>\p{General_Category=Open_Punctuation}</code>) (72)
X <code>\p{PUA}</code>	<code>\p{Private_Use_Area}</code> (= <code>\p{Block=Private_Use_Area}</code>) (6400)
<code>\p{Punct}</code>	<code>\p{General_Category=Punctuation}</code> (Short: <code>\p{P}</code> ; NOT <code>\p{General_Punctuation}</code>) (632)
<code>\p{Punctuation}</code>	<code>\p{Punct}</code> (= <code>\p{General_Category=Punctuation}</code>) (NOT <code>\p{General_Punctuation}</code>) (632)
<code>\p{Qaac}</code>	<code>\p{Coptic}</code> (= <code>\p{Script=Coptic}</code>) (NOT <code>\p{Block=Coptic}</code>) (137)
<code>\p{Qaai}</code>	<code>\p{Inherited}</code> (= <code>\p{Script=Inherited}</code>) (523)
<code>\p{QMark}</code>	<code>\p{Quotation_Mark}</code> (= <code>\p{Quotation_Mark=Y}</code>) (29)
<code>\p{QMark: *}</code>	<code>\p{Quotation_Mark: *}</code>
<code>\p{Quotation_Mark}</code>	<code>\p{Quotation_Mark=Y}</code> (Short: <code>\p{QMark}</code>) (29)
<code>\p{Quotation_Mark: N*}</code>	(Short: <code>\p{QMark=N}</code> , <code>\p{QMark}</code>) (1_114_083)
<code>\p{Quotation_Mark: Y*}</code>	(Short: <code>\p{QMark=Y}</code> , <code>\p{QMark}</code>) (29)
<code>\p{Radical}</code>	<code>\p{Radical=Y}</code> (329)
<code>\p{Radical: N*}</code>	(Single: <code>\p{Radical}</code>) (1_113_783)
<code>\p{Radical: Y*}</code>	(Single: <code>\p{Radical}</code>) (329)
<code>\p{Rejang}</code>	<code>\p{Script=Rejang}</code> (Short: <code>\p{Rjng}</code> ; NOT <code>\p{Block=Rejang}</code>) (37)
<code>\p{Rjng}</code>	<code>\p{Rejang}</code> (= <code>\p{Script=Rejang}</code>) (NOT <code>\p{Block=Rejang}</code>) (37)
X <code>\p{Rumi}</code>	<code>\p{Rumi_Numeral_Symbols}</code> (= <code>\p{Block=Rumi_Numeral_Symbols}</code>) (32)
X <code>\p{Rumi_Numeral_Symbols}</code>	<code>\p{Block=Rumi_Numeral_Symbols}</code> (Short: <code>\p{InRumi}</code>) (32)
<code>\p{Runic}</code>	<code>\p{Script=Runic}</code> (Short: <code>\p{Runr}</code> ; NOT <code>\p{Block=Runic}</code>) (78)
<code>\p{Runr}</code>	<code>\p{Runic}</code> (= <code>\p{Script=Runic}</code>) (NOT <code>\p{Block=Runic}</code>) (78)
<code>\p{S}</code>	<code>\p{Symbol}</code> (= <code>\p{General_Category=Symbol}</code>) (5520)
<code>\p{Samaritan}</code>	<code>\p{Script=Samaritan}</code> (Short: <code>\p{Samr}</code> ; NOT <code>\p{Block=Samaritan}</code>) (61)
<code>\p{Samr}</code>	<code>\p{Samaritan}</code> (= <code>\p{Script=Samaritan}</code>) (NOT <code>\p{Block=Samaritan}</code>) (61)
<code>\p{Sarab}</code>	<code>\p{Old_South_Arabian}</code> (= <code>\p{Script=Old_South_Arabian}</code>) (32)
<code>\p{Saur}</code>	<code>\p{Saurashtra}</code> (= <code>\p{Script=Saurashtra}</code>) (NOT <code>\p{Block=Saurashtra}</code>) (81)

<code>\p{Saurashtra}</code>	<code>\p{Script=Saurashtra}</code> (Short: <code>\p{Saur}</code> ; NOT <code>\p{Block=Saurashtra}</code>) (81)
<code>\p{SB: *}</code>	<code>\p{Sentence_Break: *}</code>
<code>\p{Sc}</code>	<code>\p{Currency_Symbol}</code> (= <code>\p{General_Category=Currency_Symbol}</code>) (49)
<code>\p{Sc: *}</code>	<code>\p{Script: *}</code>
<code>\p{Script: Arab}</code>	<code>\p{Script=Arabic}</code> (1235)
<code>\p{Script: Arabic}</code>	(Short: <code>\p{Sc=Arab}</code> , <code>\p{Arab}</code>) (1235)
<code>\p{Script: Armenian}</code>	(Short: <code>\p{Sc=Armn}</code> , <code>\p{Armn}</code>) (91)
<code>\p{Script: Armi}</code>	<code>\p{Script=Imperial_Aramaic}</code> (31)
<code>\p{Script: Armn}</code>	<code>\p{Script=Armenian}</code> (91)
<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: Cakm}</code>	<code>\p{Script=Chakma}</code> (67)
<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: Chakma}</code>	(Short: <code>\p{Sc=Cakm}</code> , <code>\p{Cakm}</code>) (67)
<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>) (6413)
<code>\p{Script: Copt}</code>	<code>\p{Script=Coptic}</code> (137)
<code>\p{Script: Coptic}</code>	(Short: <code>\p{Sc=Copt}</code> , <code>\p{Copt}</code>) (137)
<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>) (417)
<code>\p{Script: Cyrl}</code>	<code>\p{Script=Cyrillic}</code> (417)
<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> (151)
<code>\p{Script: Devanagari}</code>	(Short: <code>\p{Sc=Deva}</code> , <code>\p{Deva}</code>) (151)
<code>\p{Script: Dsrt}</code>	<code>\p{Script=Deseret}</code> (80)
<code>\p{Script: Eryp}</code>	<code>\p{Script=Egyptian_Hieroglyphs}</code> (1071)
<code>\p{Script: Egyptian_Hieroglyphs}</code>	(Short: <code>\p{Sc=Eryp}</code> , <code>\p{Eryp}</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> (127)
<code>\p{Script: Georgian}</code>	(Short: <code>\p{Sc=Geor}</code> , <code>\p{Geor}</code>) (127)
<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>) (84)
<code>\p{Script: Gujr}</code>	<code>\p{Script=Gujarati}</code> (84)
<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_963)
<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_963)
<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>) (67)
<code>\p{Script: Laoo}</code>	<code>\p{Script=Lao}</code> (67)
<code>\p{Script: Latin}</code>	(Short: <code>\p{Sc=Latn}</code> , <code>\p{Latn}</code>) (1272)
<code>\p{Script: Latn}</code>	<code>\p{Script=Latin}</code> (1272)
<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>) (79)
<code>\p{Script: Merc}</code>	<code>\p{Script=Meroitic_Cursive}</code> (26)
<code>\p{Script: Mero}</code>	<code>\p{Script=Meroitic_Hieroglyphs}</code> (32)
<code>\p{Script: Meroitic_Cursive}</code>	(Short: <code>\p{Sc=Merc}</code> , <code>\p{Merc}</code>) (26)
<code>\p{Script: Meroitic_Hieroglyphs}</code>	(Short: <code>\p{Sc=Mero}</code> , <code>\p{Mero}</code>) (32)
<code>\p{Script: Miao}</code>	(Short: <code>\p{Sc=Miao}</code> , <code>\p{Miao}</code>) (133)
<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> (79)
<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: Plrd}</code>	<code>\p{Script=Miao}</code> (133)
<code>\p{Script: Prti}</code>	<code>\p{Script=Inscriptional_Parthian}</code> (30)
<code>\p{Script: Qaac}</code>	<code>\p{Script=Coptic}</code> (137)
<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: Sharada}</code>	(Short: <code>\p{Sc=Shrd}</code> , <code>\p{Shrd}</code>) (83)
<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: Shrd}</code>	<code>\p{Script=Sharada}</code> (83)
<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: Sora}</code>	<code>\p{Script=Sora_Sompeng}</code> (35)
<code>\p{Script: Sora_Sompeng}</code>	(Short: <code>\p{Sc=Sora}</code> , <code>\p{Sora}</code>) (35)
<code>\p{Script: Sund}</code>	<code>\p{Script=Sundanese}</code> (72)
<code>\p{Script: Sundanese}</code>	(Short: <code>\p{Sc=Sund}</code> , <code>\p{Sund}</code>) (72)
<code>\p{Script: Sylo}</code>	<code>\p{Script=Syloti_Nagri}</code> (44)
<code>\p{Script: Syloti_Nagri}</code>	(Short: <code>\p{Sc=Sylo}</code> , <code>\p{Sylo}</code>) (44)
<code>\p{Script: Syrc}</code>	<code>\p{Script=Syriac}</code> (77)
<code>\p{Script: Syriac}</code>	(Short: <code>\p{Sc=Syrc}</code> , <code>\p{Syrc}</code>) (77)
<code>\p{Script: Tagalog}</code>	(Short: <code>\p{Sc=Tglg}</code> , <code>\p{Tglg}</code>) (20)
<code>\p{Script: Tagb}</code>	<code>\p{Script=Tagbanwa}</code> (18)
<code>\p{Script: Tagbanwa}</code>	(Short: <code>\p{Sc=Tagb}</code> , <code>\p{Tagb}</code>) (18)
<code>\p{Script: Tai_Le}</code>	(Short: <code>\p{Sc=Tale}</code> , <code>\p{Tale}</code>) (35)
<code>\p{Script: Tai_Tham}</code>	(Short: <code>\p{Sc=Lana}</code> , <code>\p{Lana}</code>) (127)
<code>\p{Script: Tai_Viet}</code>	(Short: <code>\p{Sc=Tavt}</code> , <code>\p{Tavt}</code>) (72)
<code>\p{Script: Takr}</code>	<code>\p{Script=Takri}</code> (66)
<code>\p{Script: Takri}</code>	(Short: <code>\p{Sc=Takr}</code> , <code>\p{Takr}</code>) (66)
<code>\p{Script: Tale}</code>	<code>\p{Script=Tai_Le}</code> (35)
<code>\p{Script: Talu}</code>	<code>\p{Script=New_Tai_Lue}</code> (83)
<code>\p{Script: Tamil}</code>	(Short: <code>\p{Sc=Taml}</code> , <code>\p{Taml}</code>) (72)
<code>\p{Script: Taml}</code>	<code>\p{Script=Tamil}</code> (72)
<code>\p{Script: Tavt}</code>	<code>\p{Script=Tai_Viet}</code> (72)
<code>\p{Script: Telu}</code>	<code>\p{Script=Telugu}</code> (93)
<code>\p{Script: Telugu}</code>	(Short: <code>\p{Sc=Telu}</code> , <code>\p{Telu}</code>) (93)
<code>\p{Script: Tfng}</code>	<code>\p{Script=Tifinagh}</code> (59)
<code>\p{Script: Tglg}</code>	<code>\p{Script=Tagalog}</code> (20)
<code>\p{Script: Thaa}</code>	<code>\p{Script=Thaana}</code> (50)
<code>\p{Script: Thaana}</code>	(Short: <code>\p{Sc=Thaa}</code> , <code>\p{Thaa}</code>) (50)
<code>\p{Script: Thai}</code>	(Short: <code>\p{Sc=Thai}</code> , <code>\p{Thai}</code>) (86)
<code>\p{Script: Tibetan}</code>	(Short: <code>\p{Sc=Tibt}</code> , <code>\p{Tibt}</code>) (207)
<code>\p{Script: Tibt}</code>	<code>\p{Script=Tibetan}</code> (207)
<code>\p{Script: Tifinagh}</code>	(Short: <code>\p{Sc=Tfng}</code> , <code>\p{Tfng}</code>) (59)
<code>\p{Script: Ugar}</code>	<code>\p{Script=Ugaritic}</code> (31)
<code>\p{Script: Ugaritic}</code>	(Short: <code>\p{Sc=Ugar}</code> , <code>\p{Ugar}</code>) (31)
<code>\p{Script: Unknown}</code>	(Short: <code>\p{Sc=Zzzz}</code> , <code>\p{Zzzz}</code>) (1_003_930)
<code>\p{Script: Vai}</code>	(Short: <code>\p{Sc=Vai}</code> , <code>\p{Vai}</code>) (300)
<code>\p{Script: Vaii}</code>	<code>\p{Script=Vai}</code> (300)
<code>\p{Script: Xpeo}</code>	<code>\p{Script=Old_Persian}</code> (50)
<code>\p{Script: Xsux}</code>	<code>\p{Script=Cuneiform}</code> (982)
<code>\p{Script: Yi}</code>	(Short: <code>\p{Sc=Yi}</code> , <code>\p{Yi}</code>) (1220)
<code>\p{Script: Yiii}</code>	<code>\p{Script=Yi}</code> (1220)
<code>\p{Script: Zinh}</code>	<code>\p{Script=Inherited}</code> (523)
<code>\p{Script: Zyyy}</code>	<code>\p{Script=Common}</code> (6413)
<code>\p{Script: Zzzz}</code>	<code>\p{Script=Unknown}</code> (1_003_930)
<code>\p{Script_Extensions: Arab}</code>	<code>\p{Script_Extensions=Arabic}</code> (1262)
<code>\p{Script_Extensions: Arabic}</code>	(Short: <code>\p{Scx=Arab}</code>) (1262)
<code>\p{Script_Extensions: Armenian}</code>	(Short: <code>\p{Scx=Armn}</code>) (92)
<code>\p{Script_Extensions: Armi}</code>	<code>\p{Script_Extensions=Imperial_Aramaic}</code> (31)
<code>\p{Script_Extensions: Armn}</code>	<code>\p{Script_Extensions=Armenian}</code> (92)
<code>\p{Script_Extensions: Avestan}</code>	(Short: <code>\p{Scx=Avst}</code>) (61)

`\p{Script_Extensions: Avst}` `\p{Script_Extensions=Avestan}` (61)
`\p{Script_Extensions: Bali}` `\p{Script_Extensions=Balinese}` (121)
`\p{Script_Extensions: Balinese}` (Short: `\p{Scx=Bali}`) (121)
`\p{Script_Extensions: Bamu}` `\p{Script_Extensions=Bamum}` (657)
`\p{Script_Extensions: Bamum}` (Short: `\p{Scx=Bamu}`) (657)
`\p{Script_Extensions: Batak}` (Short: `\p{Scx=Batk}`) (56)
`\p{Script_Extensions: Batk}` `\p{Script_Extensions=Batak}` (56)
`\p{Script_Extensions: Beng}` `\p{Script_Extensions=Bengali}` (94)
`\p{Script_Extensions: Bengali}` (Short: `\p{Scx=Beng}`) (94)
`\p{Script_Extensions: Bopo}` `\p{Script_Extensions=Bopomofo}` (306)
`\p{Script_Extensions: Bopomofo}` (Short: `\p{Scx=Bopo}`) (306)
`\p{Script_Extensions: Brah}` `\p{Script_Extensions=Brahmi}` (108)
`\p{Script_Extensions: Brahmi}` (Short: `\p{Scx=Brah}`) (108)
`\p{Script_Extensions: Brai}` `\p{Script_Extensions=Braille}` (256)
`\p{Script_Extensions: Braille}` (Short: `\p{Scx=Brai}`) (256)
`\p{Script_Extensions: Bugi}` `\p{Script_Extensions=Buginese}` (30)
`\p{Script_Extensions: Buginese}` (Short: `\p{Scx=Bugi}`) (30)
`\p{Script_Extensions: Buhd}` `\p{Script_Extensions=Buhid}` (22)
`\p{Script_Extensions: Buhid}` (Short: `\p{Scx=Buhd}`) (22)
`\p{Script_Extensions: Cakm}` `\p{Script_Extensions=Chakma}` (67)
`\p{Script_Extensions: Canadian_Aboriginal}` (Short: `\p{Scx=Cans}`)
(710)
`\p{Script_Extensions: Cans}` `\p{Script_Extensions=`
`Canadian_Aboriginal}` (710)
`\p{Script_Extensions: Cari}` `\p{Script_Extensions=Carian}` (49)
`\p{Script_Extensions: Carian}` (Short: `\p{Scx=Cari}`) (49)
`\p{Script_Extensions: Chakma}` (Short: `\p{Scx=Cakm}`) (67)
`\p{Script_Extensions: Cham}` (Short: `\p{Scx=Cham}`) (83)
`\p{Script_Extensions: Cher}` `\p{Script_Extensions=Cherokee}` (85)
`\p{Script_Extensions: Cherokee}` (Short: `\p{Scx=Cher}`) (85)
`\p{Script_Extensions: Common}` (Short: `\p{Scx=Zyyy}`) (6057)
`\p{Script_Extensions: Copt}` `\p{Script_Extensions=Coptic}` (137)
`\p{Script_Extensions: Coptic}` (Short: `\p{Scx=Copt}`) (137)
`\p{Script_Extensions: Cprt}` `\p{Script_Extensions=Cypriot}` (112)
`\p{Script_Extensions: Cuneiform}` (Short: `\p{Scx=Xsux}`) (982)
`\p{Script_Extensions: Cypriot}` (Short: `\p{Scx=Cprt}`) (112)
`\p{Script_Extensions: Cyrillic}` (Short: `\p{Scx=Cyrl}`) (419)
`\p{Script_Extensions: Cyril}` `\p{Script_Extensions=Cyrillic}` (419)
`\p{Script_Extensions: Deseret}` (Short: `\p{Scx=Dsrt}`) (80)
`\p{Script_Extensions: Deva}` `\p{Script_Extensions=Devanagari}` (193)
`\p{Script_Extensions: Devanagari}` (Short: `\p{Scx=Deva}`) (193)
`\p{Script_Extensions: Dsrt}` `\p{Script_Extensions=Deseret}` (80)
`\p{Script_Extensions: Egyp}` `\p{Script_Extensions=`
`Egyptian_Hieroglyphs}` (1071)
`\p{Script_Extensions: Egyptian_Hieroglyphs}` (Short: `\p{Scx=Egyp}`)
(1071)
`\p{Script_Extensions: Ethi}` `\p{Script_Extensions=Ethiopic}` (495)
`\p{Script_Extensions: Ethiopic}` (Short: `\p{Scx=Ethi}`) (495)
`\p{Script_Extensions: Geor}` `\p{Script_Extensions=Georgian}` (128)
`\p{Script_Extensions: Georgian}` (Short: `\p{Scx=Geor}`) (128)
`\p{Script_Extensions: Glag}` `\p{Script_Extensions=Glagolitic}` (94)
`\p{Script_Extensions: Glagolitic}` (Short: `\p{Scx=Glag}`) (94)
`\p{Script_Extensions: Goth}` `\p{Script_Extensions=Gothic}` (27)
`\p{Script_Extensions: Gothic}` (Short: `\p{Scx=Goth}`) (27)
`\p{Script_Extensions: Greek}` (Short: `\p{Scx=Grek}`) (515)
`\p{Script_Extensions: Grek}` `\p{Script_Extensions=Greek}` (515)

`\p{Script_Extensions: Gujarati}` (Short: `\p{Scx=Gujr}`) (94)
`\p{Script_Extensions: Gujr}` `\p{Script_Extensions=Gujarati}` (94)
`\p{Script_Extensions: Gurmukhi}` (Short: `\p{Scx=Guru}`) (91)
`\p{Script_Extensions: Guru}` `\p{Script_Extensions=Gurmukhi}` (91)
`\p{Script_Extensions: Han}` (Short: `\p{Scx=Han}`) (76_218)
`\p{Script_Extensions: Hang}` `\p{Script_Extensions=Hangul}` (11_971)
`\p{Script_Extensions: Hangul}` (Short: `\p{Scx=Hang}`) (11_971)
`\p{Script_Extensions: Hani}` `\p{Script_Extensions=Han}` (76_218)
`\p{Script_Extensions: Hano}` `\p{Script_Extensions=Hanunoo}` (23)
`\p{Script_Extensions: Hanunoo}` (Short: `\p{Scx=Hano}`) (23)
`\p{Script_Extensions: Hebr}` `\p{Script_Extensions=Hebrew}` (133)
`\p{Script_Extensions: Hebrew}` (Short: `\p{Scx=Hebr}`) (133)
`\p{Script_Extensions: Hira}` `\p{Script_Extensions=Hiragana}` (356)
`\p{Script_Extensions: Hiragana}` (Short: `\p{Scx=Hira}`) (356)
`\p{Script_Extensions: Imperial_Aramaic}` (Short: `\p{Scx=Armi}`) (31)
`\p{Script_Extensions: Inherited}` (Short: `\p{Scx=Zinh}`) (459)
`\p{Script_Extensions: Inscriptional_Pahlavi}` (Short: `\p{Scx=Phli}`)
(27)
`\p{Script_Extensions: Inscriptional_Parthian}` (Short: `\p{Scx=`
`Prti}`) (30)
`\p{Script_Extensions: Ital}` `\p{Script_Extensions=Old_Italic}` (35)
`\p{Script_Extensions: Java}` `\p{Script_Extensions=Javanese}` (91)
`\p{Script_Extensions: Javanese}` (Short: `\p{Scx=Java}`) (91)
`\p{Script_Extensions: Kaithi}` (Short: `\p{Scx=Kthi}`) (76)
`\p{Script_Extensions: Kali}` `\p{Script_Extensions=Kayah_Li}` (48)
`\p{Script_Extensions: Kana}` `\p{Script_Extensions=Katakana}` (565)
`\p{Script_Extensions: Kannada}` (Short: `\p{Scx=Knda}`) (86)
`\p{Script_Extensions: Katakana}` (Short: `\p{Scx=Kana}`) (565)
`\p{Script_Extensions: Kayah_Li}` (Short: `\p{Scx=Kali}`) (48)
`\p{Script_Extensions: Khar}` `\p{Script_Extensions=Kharoshthi}` (65)
`\p{Script_Extensions: Kharoshthi}` (Short: `\p{Scx=Khar}`) (65)
`\p{Script_Extensions: Khmer}` (Short: `\p{Scx=Khmr}`) (146)
`\p{Script_Extensions: Khmr}` `\p{Script_Extensions=Khmer}` (146)
`\p{Script_Extensions: Knda}` `\p{Script_Extensions=Kannada}` (86)
`\p{Script_Extensions: Kthi}` `\p{Script_Extensions=Kaithi}` (76)
`\p{Script_Extensions: Lana}` `\p{Script_Extensions=Tai_Tham}` (127)
`\p{Script_Extensions: Lao}` (Short: `\p{Scx=Lao}`) (67)
`\p{Script_Extensions: Laoo}` `\p{Script_Extensions=Lao}` (67)
`\p{Script_Extensions: Latin}` (Short: `\p{Scx=Latn}`) (1289)
`\p{Script_Extensions: Latn}` `\p{Script_Extensions=Latin}` (1289)
`\p{Script_Extensions: Lepc}` `\p{Script_Extensions=Lepcha}` (74)
`\p{Script_Extensions: Lepcha}` (Short: `\p{Scx=Lepc}`) (74)
`\p{Script_Extensions: Limb}` `\p{Script_Extensions=Limbu}` (66)
`\p{Script_Extensions: Limbu}` (Short: `\p{Scx=Limb}`) (66)
`\p{Script_Extensions: Linb}` `\p{Script_Extensions=Linear_B}` (268)
`\p{Script_Extensions: Linear_B}` (Short: `\p{Scx=Linb}`) (268)
`\p{Script_Extensions: Lisu}` (Short: `\p{Scx=Lisu}`) (48)
`\p{Script_Extensions: Lyci}` `\p{Script_Extensions=Lycian}` (29)
`\p{Script_Extensions: Lycian}` (Short: `\p{Scx=Lyci}`) (29)
`\p{Script_Extensions: Lydi}` `\p{Script_Extensions=Lydian}` (27)
`\p{Script_Extensions: Lydian}` (Short: `\p{Scx=Lydi}`) (27)
`\p{Script_Extensions: Malayalam}` (Short: `\p{Scx=Mlym}`) (98)
`\p{Script_Extensions: Mand}` `\p{Script_Extensions=Mandaic}` (30)
`\p{Script_Extensions: Mandaic}` (Short: `\p{Scx=Mand}`) (30)
`\p{Script_Extensions: Meetei_Mayek}` (Short: `\p{Scx=Mtei}`) (79)
`\p{Script_Extensions: Merc}` `\p{Script_Extensions=Meroitic_Cursive}`

```

(26)
\p{Script_Extensions: Mero} \p{Script_Extensions=
    Meroitic_Hieroglyphs} (32)
\p{Script_Extensions: Meroitic_Cursive} (Short: \p{Scx=Merc}) (26)
\p{Script_Extensions: Meroitic_Hieroglyphs} (Short: \p{Scx=Mero})
    (32)
\p{Script_Extensions: Miao} (Short: \p{Scx=Miao}) (133)
\p{Script_Extensions: Mlym} \p{Script_Extensions=Malayalam} (98)
\p{Script_Extensions: Mong} \p{Script_Extensions=Mongolian} (156)
\p{Script_Extensions: Mongolian} (Short: \p{Scx=Mong}) (156)
\p{Script_Extensions: Mtei} \p{Script_Extensions=Meetei_Mayek} (79)
\p{Script_Extensions: Myanmar} (Short: \p{Scx=Mymr}) (188)
\p{Script_Extensions: Mymr} \p{Script_Extensions=Myanmar} (188)
\p{Script_Extensions: New_Tai_Lue} (Short: \p{Scx=Talु}) (83)
\p{Script_Extensions: Nko} (Short: \p{Scx=Nko}) (59)
\p{Script_Extensions: Nkoo} \p{Script_Extensions=Nko} (59)
\p{Script_Extensions: Ogam} \p{Script_Extensions=Ogham} (29)
\p{Script_Extensions: Ogham} (Short: \p{Scx=Ogam}) (29)
\p{Script_Extensions: Ol_Chiki} (Short: \p{Scx=Olck}) (48)
\p{Script_Extensions: Olck} \p{Script_Extensions=Ol_Chiki} (48)
\p{Script_Extensions: Old_Italic} (Short: \p{Scx=Ital}) (35)
\p{Script_Extensions: Old_Persian} (Short: \p{Scx=Xpeo}) (50)
\p{Script_Extensions: Old_South_Arabian} (Short: \p{Scx=Sarb}) (32)
\p{Script_Extensions: Old_Turkic} (Short: \p{Scx=Orkh}) (73)
\p{Script_Extensions: Oriya} (Short: \p{Scx=Orya}) (92)
\p{Script_Extensions: Orkh} \p{Script_Extensions=Old_Turkic} (73)
\p{Script_Extensions: Orya} \p{Script_Extensions=Oriya} (92)
\p{Script_Extensions: Osma} \p{Script_Extensions=Osmanya} (40)
\p{Script_Extensions: Osmanya} (Short: \p{Scx=Osma}) (40)
\p{Script_Extensions: Phag} \p{Script_Extensions=Phags_Pa} (59)
\p{Script_Extensions: Phags_Pa} (Short: \p{Scx=Phag}) (59)
\p{Script_Extensions: Phli} \p{Script_Extensions=
    Inscriptional_Pahlavi} (27)
\p{Script_Extensions: Phnx} \p{Script_Extensions=Phoenician} (29)
\p{Script_Extensions: Phoenician} (Short: \p{Scx=Phnx}) (29)
\p{Script_Extensions: Plrd} \p{Script_Extensions=Miao} (133)
\p{Script_Extensions: Prti} \p{Script_Extensions=
    Inscriptional_Parthian} (30)
\p{Script_Extensions: Qaac} \p{Script_Extensions=Coptic} (137)
\p{Script_Extensions: Qaai} \p{Script_Extensions=Inherited} (459)
\p{Script_Extensions: Rejang} (Short: \p{Scx=Rjng}) (37)
\p{Script_Extensions: Rjng} \p{Script_Extensions=Rejang} (37)
\p{Script_Extensions: Runic} (Short: \p{Scx=Runr}) (78)
\p{Script_Extensions: Runr} \p{Script_Extensions=Runic} (78)
\p{Script_Extensions: Samaritan} (Short: \p{Scx=Samr}) (61)
\p{Script_Extensions: Samr} \p{Script_Extensions=Samaritan} (61)
\p{Script_Extensions: Sarb} \p{Script_Extensions=
    Old_South_Arabian} (32)
\p{Script_Extensions: Saur} \p{Script_Extensions=Saurashtra} (81)
\p{Script_Extensions: Saurashtra} (Short: \p{Scx=Saur}) (81)
\p{Script_Extensions: Sharada} (Short: \p{Scx=Shrd}) (83)
\p{Script_Extensions: Shavian} (Short: \p{Scx=Shaw}) (48)
\p{Script_Extensions: Shaw} \p{Script_Extensions=Shavian} (48)
\p{Script_Extensions: Shrd} \p{Script_Extensions=Sharada} (83)
\p{Script_Extensions: Sinh} \p{Script_Extensions=Sinhala} (80)
\p{Script_Extensions: Sinhala} (Short: \p{Scx=Sinh}) (80)

```

```

\p{Script_Extensions: Sora} \p{Script_Extensions=Sora_Sompeng} (35)
\p{Script_Extensions: Sora_Sompeng} (Short: \p{Scx=Sora}) (35)
\p{Script_Extensions: Sund} \p{Script_Extensions=Sundanese} (72)
\p{Script_Extensions: Sundanese} (Short: \p{Scx=Sund}) (72)
\p{Script_Extensions: Sylo} \p{Script_Extensions=Syloti_Nagri} (44)
\p{Script_Extensions: Syloti_Nagri} (Short: \p{Scx=Sylo}) (44)
\p{Script_Extensions: Syrc} \p{Script_Extensions=Syriac} (93)
\p{Script_Extensions: Syriac} (Short: \p{Scx=Syrc}) (93)
\p{Script_Extensions: Tagalog} (Short: \p{Scx=Tglg}) (22)
\p{Script_Extensions: Tagb} \p{Script_Extensions=Tagbanwa} (20)
\p{Script_Extensions: Tagbanwa} (Short: \p{Scx=Tagb}) (20)
\p{Script_Extensions: Tai_Le} (Short: \p{Scx=Tale}) (35)
\p{Script_Extensions: Tai_Tham} (Short: \p{Scx=Lana}) (127)
\p{Script_Extensions: Tai_Viet} (Short: \p{Scx=Tavt}) (72)
\p{Script_Extensions: Takr} \p{Script_Extensions=Takri} (78)
\p{Script_Extensions: Takri} (Short: \p{Scx=Takr}) (78)
\p{Script_Extensions: Tale} \p{Script_Extensions=Tai_Le} (35)
\p{Script_Extensions: Talu} \p{Script_Extensions=New_Tai_Lue} (83)
\p{Script_Extensions: Tamil} (Short: \p{Scx=Taml}) (72)
\p{Script_Extensions: Taml} \p{Script_Extensions=Tamil} (72)
\p{Script_Extensions: Tavt} \p{Script_Extensions=Tai_Viet} (72)
\p{Script_Extensions: Telu} \p{Script_Extensions=Telugu} (93)
\p{Script_Extensions: Telugu} (Short: \p{Scx=Telu}) (93)
\p{Script_Extensions: Tfng} \p{Script_Extensions=Tifinagh} (59)
\p{Script_Extensions: Tglg} \p{Script_Extensions=Tagalog} (22)
\p{Script_Extensions: Thaa} \p{Script_Extensions=Thaana} (65)
\p{Script_Extensions: Thaana} (Short: \p{Scx=Thaa}) (65)
\p{Script_Extensions: Thai} (Short: \p{Scx=Thai}) (86)
\p{Script_Extensions: Tibetan} (Short: \p{Scx=Tibt}) (207)
\p{Script_Extensions: Tibt} \p{Script_Extensions=Tibetan} (207)
\p{Script_Extensions: Tifinagh} (Short: \p{Scx=Tfng}) (59)
\p{Script_Extensions: Ugar} \p{Script_Extensions=Ugaritic} (31)
\p{Script_Extensions: Ugaritic} (Short: \p{Scx=Ugar}) (31)
\p{Script_Extensions: Unknown} (Short: \p{Scx=Zzzz}) (1_003_930)
\p{Script_Extensions: Vai} (Short: \p{Scx=Vai}) (300)
\p{Script_Extensions: Vaih} \p{Script_Extensions=Vai} (300)
\p{Script_Extensions: Xpeo} \p{Script_Extensions=Old_Persian} (50)
\p{Script_Extensions: Xsux} \p{Script_Extensions=Cuneiform} (982)
\p{Script_Extensions: Yi} (Short: \p{Scx=Yi}) (1246)
\p{Script_Extensions: Yih} \p{Script_Extensions=Yi} (1246)
\p{Script_Extensions: Zinh} \p{Script_Extensions=Inherited} (459)
\p{Script_Extensions: Zyyy} \p{Script_Extensions=Common} (6057)
\p{Script_Extensions: Zzzz} \p{Script_Extensions=Unknown}
(1_003_930)

\p{Scx: *} \p{Script_Extensions: *}
\p{SD} \p{Soft_Dotted} (= \p{Soft_Dotted=Y}) (46)
\p{SD: *} \p{Soft_Dotted: *}
\p{Sentence_Break: AT} \p{Sentence_Break=ATerm} (4)
\p{Sentence_Break: ATerm} (Short: \p{SB=AT}) (4)
\p{Sentence_Break: CL} \p{Sentence_Break=Close} (177)
\p{Sentence_Break: Close} (Short: \p{SB=CL}) (177)
\p{Sentence_Break: CR} (Short: \p{SB=CR}) (1)
\p{Sentence_Break: EX} \p{Sentence_Break=Extend} (1649)
\p{Sentence_Break: Extend} (Short: \p{SB=EX}) (1649)
\p{Sentence_Break: FO} \p{Sentence_Break=Format} (137)
\p{Sentence_Break: Format} (Short: \p{SB=FO}) (137)

```

<code>\p{Sentence_Break: LE}</code>	<code>\p{Sentence_Break=OLetter}</code> (97_841)
<code>\p{Sentence_Break: LF}</code>	(Short: <code>\p{SB=LF}</code>) (1)
<code>\p{Sentence_Break: LO}</code>	<code>\p{Sentence_Break=Lower}</code> (1933)
<code>\p{Sentence_Break: Lower}</code>	(Short: <code>\p{SB=LO}</code>) (1933)
<code>\p{Sentence_Break: NU}</code>	<code>\p{Sentence_Break=Numeric}</code> (452)
<code>\p{Sentence_Break: Numeric}</code>	(Short: <code>\p{SB=NU}</code>) (452)
<code>\p{Sentence_Break: OLetter}</code>	(Short: <code>\p{SB=LE}</code>) (97_841)
<code>\p{Sentence_Break: Other}</code>	(Short: <code>\p{SB=XX}</code>) (1_010_273)
<code>\p{Sentence_Break: SC}</code>	<code>\p{Sentence_Break=SContinue}</code> (26)
<code>\p{Sentence_Break: SContinue}</code>	(Short: <code>\p{SB=SC}</code>) (26)
<code>\p{Sentence_Break: SE}</code>	<code>\p{Sentence_Break=Sep}</code> (3)
<code>\p{Sentence_Break: Sep}</code>	(Short: <code>\p{SB=SE}</code>) (3)
<code>\p{Sentence_Break: Sp}</code>	(Short: <code>\p{SB=Sp}</code>) (21)
<code>\p{Sentence_Break: ST}</code>	<code>\p{Sentence_Break=STerm}</code> (80)
<code>\p{Sentence_Break: STerm}</code>	(Short: <code>\p{SB=ST}</code>) (80)
<code>\p{Sentence_Break: UP}</code>	<code>\p{Sentence_Break=Upper}</code> (1514)
<code>\p{Sentence_Break: Upper}</code>	(Short: <code>\p{SB=UP}</code>) (1514)
<code>\p{Sentence_Break: XX}</code>	<code>\p{Sentence_Break=Other}</code> (1_010_273)
<code>\p{Separator}</code>	<code>\p{General_Category=Separator}</code> (Short: <code>\p{Z}</code>) (20)
<code>\p{Sharada}</code>	<code>\p{Script=Sharada}</code> (Short: <code>\p{Shrd}</code> ; NOT <code>\p{Block=Sharada}</code>) (83)
<code>\p{Shavian}</code>	<code>\p{Script=Shavian}</code> (Short: <code>\p{Shaw}</code>) (48)
<code>\p{Shaw}</code>	<code>\p{Shavian}</code> (= <code>\p{Script=Shavian}</code>) (48)
<code>\p{Shrd}</code>	<code>\p{Sharada}</code> (= <code>\p{Script=Sharada}</code>) (NOT <code>\p{Block=Sharada}</code>) (83)
<code>\p{Sinh}</code>	<code>\p{Sinhala}</code> (= <code>\p{Script=Sinhala}</code>) (NOT <code>\p{Block=Sinhala}</code>) (80)
<code>\p{Sinhala}</code>	<code>\p{Script=Sinhala}</code> (Short: <code>\p{Sinh}</code> ; NOT <code>\p{Block=Sinhala}</code>) (80)
<code>\p{Sk}</code>	<code>\p{Modifier_Symbol}</code> (= <code>\p{General_Category=Modifier_Symbol}</code>) (115)
<code>\p{Sm}</code>	<code>\p{Math_Symbol}</code> (= <code>\p{General_Category=Math_Symbol}</code>) (952)
X <code>\p{Small_Form_Variants}</code>	<code>\p{Block=Small_Form_Variants}</code> (Short: <code>\p{InSmallForms}</code>) (32)
X <code>\p{Small_Forms}</code>	<code>\p{Small_Form_Variants}</code> (= <code>\p{Block=Small_Form_Variants}</code>) (32)
<code>\p{So}</code>	<code>\p{Other_Symbol}</code> (= <code>\p{General_Category=Other_Symbol}</code>) (4404)
<code>\p{Soft_Dotted}</code>	<code>\p{Soft_Dotted=Y}</code> (Short: <code>\p{SD}</code>) (46)
<code>\p{Soft_Dotted: N*}</code>	(Short: <code>\p{SD=N}</code> , <code>\p{SD}</code>) (1_114_066)
<code>\p{Soft_Dotted: Y*}</code>	(Short: <code>\p{SD=Y}</code> , <code>\p{SD}</code>) (46)
<code>\p{Sora}</code>	<code>\p{Sora_Sompeng}</code> (= <code>\p{Script=Sora_Sompeng}</code>) (NOT <code>\p{Block=Sora_Sompeng}</code>) (35)
<code>\p{Sora_Sompeng}</code>	<code>\p{Script=Sora_Sompeng}</code> (Short: <code>\p{Sora}</code> ; NOT <code>\p{Block=Sora_Sompeng}</code>) (35)
<code>\p{Space}</code>	<code>\p{White_Space=Y}</code> \s including beyond ASCII and vertical tab (26)
<code>\p{Space: *}</code>	<code>\p{White_Space: *}</code>
<code>\p{Space_Separator}</code>	<code>\p{General_Category=Space_Separator}</code> (Short: <code>\p{Zs}</code>) (18)
<code>\p{SpacePerl}</code>	<code>\p{XPerlSpace}</code> (26)
<code>\p{Spacing_Mark}</code>	<code>\p{General_Category=Spacing_Mark}</code> (Short:

	<code>\p{Mc}</code>	(353)
X <code>\p{Spacing_Modifier_Letters}</code>	<code>\p{Block=Spacing_Modifier_Letters}</code>	(Short: <code>\p{InModifierLetters}</code>) (80)
X <code>\p{Specials}</code>	<code>\p{Block=Specials}</code>	(16)
<code>\p{STerm}</code>	<code>\p{STerm=Y}</code>	(83)
<code>\p{STerm: N*}</code>	(Single: <code>\p{STerm}</code>)	(1_114_029)
<code>\p{STerm: Y*}</code>	(Single: <code>\p{STerm}</code>)	(83)
<code>\p{Sund}</code>	<code>\p{Sundanese}</code> (= <code>\p{Script=Sundanese}</code>)	(NOT <code>\p{Block=Sundanese}</code>) (72)
<code>\p{Sundanese}</code>	<code>\p{Script=Sundanese}</code> (Short: <code>\p{Sund}</code> ; NOT <code>\p{Block=Sundanese}</code>)	(72)
X <code>\p{Sundanese_Sup}</code>	<code>\p{Sundanese_Supplement}</code> (= <code>\p{Block=Sundanese_Supplement}</code>)	(16)
X <code>\p{Sundanese_Supplement}</code>	<code>\p{Block=Sundanese_Supplement}</code> (Short: <code>\p{InSundaneseSup}</code>)	(16)
X <code>\p{Sup_Arrows_A}</code>	<code>\p{Supplemental_Arrows_A}</code> (= <code>\p{Block=Supplemental_Arrows_A}</code>)	(16)
X <code>\p{Sup_Arrows_B}</code>	<code>\p{Supplemental_Arrows_B}</code> (= <code>\p{Block=Supplemental_Arrows_B}</code>)	(128)
X <code>\p{Sup_Math_Operators}</code>	<code>\p{Supplemental_Mathematical_Operators}</code> (= <code>\p{Block=Supplemental_Mathematical_Operators}</code>)	(256)
X <code>\p{Sup_PUA_A}</code>	<code>\p{Supplementary_Private_Use_Area_A}</code> (= <code>\p{Block=Supplementary_Private_Use_Area_A}</code>)	(65_536)
X <code>\p{Sup_PUA_B}</code>	<code>\p{Supplementary_Private_Use_Area_B}</code> (= <code>\p{Block=Supplementary_Private_Use_Area_B}</code>)	(65_536)
X <code>\p{Sup_Punctuation}</code>	<code>\p{Supplemental_Punctuation}</code> (= <code>\p{Block=Supplemental_Punctuation}</code>)	(128)
X <code>\p{Super_And_Sub}</code>	<code>\p{Superscripts_And_Subscripts}</code> (= <code>\p{Block=Superscripts_And_Subscripts}</code>)	(48)
X <code>\p{Superscripts_And_Subscripts}</code>	<code>\p{Block=Superscripts_And_Subscripts}</code> (Short: <code>\p{InSuperAndSub}</code>)	(48)
X <code>\p{Supplemental_Arrows_A}</code>	<code>\p{Block=Supplemental_Arrows_A}</code> (Short: <code>\p{InSupArrowsA}</code>)	(16)
X <code>\p{Supplemental_Arrows_B}</code>	<code>\p{Block=Supplemental_Arrows_B}</code> (Short: <code>\p{InSupArrowsB}</code>)	(128)
X <code>\p{Supplemental_Mathematical_Operators}</code>	<code>\p{Block=Supplemental_Mathematical_Operators}</code> (Short: <code>\p{InSupMathOperators}</code>)	(256)
X <code>\p{Supplemental_Punctuation}</code>	<code>\p{Block=Supplemental_Punctuation}</code> (Short: <code>\p{InSupPunctuation}</code>)	(128)
X <code>\p{Supplementary_Private_Use_Area_A}</code>	<code>\p{Block=Supplementary_Private_Use_Area_A}</code> (Short: <code>\p{InSupPUAA}</code>)	(65_536)
X <code>\p{Supplementary_Private_Use_Area_B}</code>	<code>\p{Block=Supplementary_Private_Use_Area_B}</code> (Short: <code>\p{InSupPUAB}</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} (= \p{Script=Syloti_Nagri} (NOT \p{Block=Syloti_Nagri})) (44)</code>
<code>\p{Syloti_Nagri}</code>	<code>\p{Script=Syloti_Nagri} (Short: \p{Sylo}; NOT \p{Block=Syloti_Nagri}) (44)</code>
<code>\p{Symbol}</code>	<code>\p{General_Category=Symbol} (Short: \p{S}) (5520)</code>
<code>\p{Syrc}</code>	<code>\p{Syriac} (= \p{Script=Syriac}) (NOT \p{Block=Syriac}) (77)</code>
<code>\p{Syriac}</code>	<code>\p{Script=Syriac} (Short: \p{Syrc}; NOT \p{Block=Syriac}) (77)</code>
<code>\p{Tagalog}</code>	<code>\p{Script=Tagalog} (Short: \p{Tglg}; NOT \p{Block=Tagalog}) (20)</code>
<code>\p{Tagb}</code>	<code>\p{Tagbanwa} (= \p{Script=Tagbanwa}) (NOT \p{Block=Tagbanwa}) (18)</code>
<code>\p{Tagbanwa}</code>	<code>\p{Script=Tagbanwa} (Short: \p{Tagb}; NOT \p{Block=Tagbanwa}) (18)</code>
X <code>\p{Tags}</code>	<code>\p{Block=Tags} (128)</code>
<code>\p{Tai_Le}</code>	<code>\p{Script=Tai_Le} (Short: \p{Tale}; NOT \p{Block=Tai_Le}) (35)</code>
<code>\p{Tai_Tham}</code>	<code>\p{Script=Tai_Tham} (Short: \p{Lana}; NOT \p{Block=Tai_Tham}) (127)</code>
<code>\p{Tai_Viet}</code>	<code>\p{Script=Tai_Viet} (Short: \p{Tavt}; NOT \p{Block=Tai_Viet}) (72)</code>
X <code>\p{Tai_Xuan_Jing}</code>	<code>\p{Tai_Xuan_Jing_Symbols} (= \p{Block=Tai_Xuan_Jing_Symbols}) (96)</code>
X <code>\p{Tai_Xuan_Jing_Symbols}</code>	<code>\p{Block=Tai_Xuan_Jing_Symbols} (Short: \p{InTaiXuanJing}) (96)</code>
<code>\p{Takr}</code>	<code>\p{Takri} (= \p{Script=Takri}) (NOT \p{Block=Takri}) (66)</code>
<code>\p{Takri}</code>	<code>\p{Script=Takri} (Short: \p{Takr}; NOT \p{Block=Takri}) (66)</code>
<code>\p{Tale}</code>	<code>\p{Tai_Le} (= \p{Script=Tai_Le}) (NOT \p{Block=Tai_Le}) (35)</code>
<code>\p{Talu}</code>	<code>\p{New_Tai_Lue} (= \p{Script=New_Tai_Lue}) (NOT \p{Block=New_Tai_Lue}) (83)</code>
<code>\p{Tamil}</code>	<code>\p{Script=Tamil} (Short: \p{Taml}; NOT \p{Block=Tamil}) (72)</code>
<code>\p{Taml}</code>	<code>\p{Tamil} (= \p{Script=Tamil}) (NOT \p{Block=Tamil}) (72)</code>
<code>\p{Tavt}</code>	<code>\p{Tai_Viet} (= \p{Script=Tai_Viet}) (NOT \p{Block=Tai_Viet}) (72)</code>
<code>\p{Telu}</code>	<code>\p{Telugu} (= \p{Script=Telugu}) (NOT \p{Block=Telugu}) (93)</code>
<code>\p{Telugu}</code>	<code>\p{Script=Telugu} (Short: \p{Telu}; NOT \p{Block=Telugu}) (93)</code>
<code>\p{Term}</code>	<code>\p{Terminal_Punctuation} (= \p{Terminal_Punctuation=Y}) (176)</code>
<code>\p{Term: *}</code>	<code>\p{Terminal_Punctuation: *}</code>
<code>\p{Terminal_Punctuation}</code>	<code>\p{Terminal_Punctuation=Y} (Short: \p{Term}) (176)</code>
<code>\p{Terminal_Punctuation: N*}</code>	<code>(Short: \p{Term=N}, \p{Term}) (1_113_936)</code>
<code>\p{Terminal_Punctuation: Y*}</code>	<code>(Short: \p{Term=Y}, \p{Term}) (176)</code>
<code>\p{Tfng}</code>	<code>\p{Tifinagh} (= \p{Script=Tifinagh}) (NOT \p{Block=Tifinagh}) (59)</code>

<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>) (59)
<code>\p{Title}</code>	<code>\p{Titlecase}</code> (/i= Cased=Yes) (31)
<code>\p{Titlecase}</code>	(= <code>\p{Gc=Lt}</code>) (Short: <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= General_Category=Cased_Letter) (31)
X <code>\p{Transport_And_Map}</code>	<code>\p{Transport_And_Map_Symbols}</code> (= <code>\p{Block=Transport_And_Map_Symbols}</code>) (128)
X <code>\p{Transport_And_Map_Symbols}</code>	<code>\p{Block=Transport_And_Map_Symbols}</code> (Short: <code>\p{InTransportAndMap}</code>) (128)
X <code>\p{UCAS}</code>	<code>\p{Unified_Canadian_Aboriginal_Syllabics}</code> (= <code>\p{Block=Unified_Canadian_Aboriginal_Syllabics}</code>) (640)
X <code>\p{UCAS_Ext}</code>	<code>\p{Unified_Canadian_Aboriginal_Syllabics_Extended}</code> (= <code>\p{Block=Unified_Canadian_Aboriginal_Syllabics_Extended}</code>) (80)
<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_617)
<code>\p{UIdeo: *}</code>	<code>\p{Unified_Ideograph: *}</code>
<code>\p{Unassigned}</code>	<code>\p{General_Category=Unassigned}</code> (Short: <code>\p{Cn}</code>) (864_414)
X <code>\p{Unified_Canadian_Aboriginal_Syllabics}</code>	<code>\p{Block=Unified_Canadian_Aboriginal_Syllabics}</code> (Short: <code>\p{InUCAS}</code>) (640)
X <code>\p{Unified_Canadian_Aboriginal_Syllabics_Extended}</code>	<code>\p{Block=Unified_Canadian_Aboriginal_Syllabics_Extended}</code> (Short: <code>\p{InUCASExt}</code>) (80)
<code>\p{Unified_Ideograph}</code>	<code>\p{Unified_Ideograph=Y}</code> (Short: <code>\p{UIdeo}</code>) (74_617)
<code>\p{Unified_Ideograph: N*}</code>	(Short: <code>\p{UIdeo=N}</code> , <code>\p{UIdeo}</code>) (1_039_495)
<code>\p{Unified_Ideograph: Y*}</code>	(Short: <code>\p{UIdeo=Y}</code> , <code>\p{UIdeo}</code>) (74_617)
<code>\p{Unknown}</code>	<code>\p{Script=Unknown}</code> (Short: <code>\p{Zzzz}</code>) (1_003_930)
<code>\p{Upper}</code>	<code>\p{Uppercase=Y}</code> (/i= Cased=Yes) (1483)
<code>\p{Upper: *}</code>	<code>\p{Uppercase: *}</code>
<code>\p{Uppercase}</code>	<code>\p{Upper}</code> (= <code>\p{Uppercase=Y}</code>) (/i= Cased=

	Yes) (1483)
<code>\p{Uppercase: N*}</code>	(Short: <code>\p{Upper=N}</code> , <code>\P{Upper}</code> ; /i= Cased=No) (1_112_629)
<code>\p{Uppercase: Y*}</code>	(Short: <code>\p{Upper=Y}</code> , <code>\P{Upper}</code> ; /i= Cased=Yes) (1483)
<code>\p{Uppercase_Letter}</code>	<code>\p{General_Category=Uppercase_Letter}</code> (Short: <code>\p{Lu}</code> ; /i= General_Category=Cased_Letter) (1441)
<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> ; NOT <code>\p{Variation_Selectors}</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> (Short: <code>\p{InVS}</code>) (16)
X <code>\p{Variation_Selectors_Supplement}</code>	<code>\p{Block=Variation_Selectors_Supplement}</code> (Short: <code>\p{InVSSup}</code>) (240)
X <code>\p{Vedic_Ext}</code>	<code>\p{Vedic_Extensions}</code> (= <code>\p{Block=Vedic_Extensions}</code>) (48)
X <code>\p{Vedic_Extensions}</code>	<code>\p{Block=Vedic_Extensions}</code> (Short: <code>\p{InVedicExt}</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>) (NOT <code>\p{Variation_Selectors}</code>) (259)
<code>\p{VS: *}</code>	<code>\p{Variation_Selector: *}</code>
X <code>\p{VS_Sup}</code>	<code>\p{Variation_Selectors_Supplement}</code> (= <code>\p{Block=Variation_Selectors_Supplement}</code>) (240)
<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> (103_406)
<code>\p{Word_Break: ALetter}</code>	(Short: <code>\p{WB=LE}</code>) (24_941)
<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>) (1649)
<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> (136)
<code>\p{Word_Break: Format}</code>	(Short: <code>\p{WB=FO}</code>) (136)
<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_941)
<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> (451)
<code>\p{Word_Break: Numeric}</code>	(Short: <code>\p{WB=NU}</code>) (451)
<code>\p{Word_Break: Other}</code>	(Short: <code>\p{WB=XX}</code>) (1_086_551)
<code>\p{Word_Break: Regional_Indicator}</code>	(Short: <code>\p{WB=RI}</code>) (26)
<code>\p{Word_Break: RI}</code>	<code>\p{Word_Break=Regional_Indicator}</code> (26)
<code>\p{Word_Break: XX}</code>	<code>\p{Word_Break=Other}</code> (1_086_551)
<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>) (103_336)
<code>\p{XID_Continue: N*}</code>	(Short: <code>\p{XIDC=N}</code> , <code>\p{XIDC}</code>) (1_010_776)
<code>\p{XID_Continue: Y*}</code>	(Short: <code>\p{XIDC=Y}</code> , <code>\p{XIDC}</code>) (103_336)
<code>\p{XID_Start}</code>	<code>\p{XID_Start=Y}</code> (Short: <code>\p{XIDS}</code>) (101_217)
<code>\p{XID_Start: N*}</code>	(Short: <code>\p{XIDS=N}</code> , <code>\p{XIDS}</code>) (1_012_895)
<code>\p{XID_Start: Y*}</code>	(Short: <code>\p{XIDS=Y}</code> , <code>\p{XIDS}</code>) (101_217)
<code>\p{XIDC}</code>	<code>\p{XID_Continue}</code> (= <code>\p{XID_Continue=Y}</code>) (103_336)
<code>\p{XIDC: *}</code>	<code>\p{XID_Continue: *}</code>
<code>\p{XIDS}</code>	<code>\p{XID_Start}</code> (= <code>\p{XID_Start=Y}</code>) (101_217)
<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>) (26)
<code>\p{XPosixAlnum}</code>	<code>\p{Alnum}</code> (102_619)
<code>\p{XPosixAlpha}</code>	<code>\p{Alpha}</code> (= <code>\p{Alphabetic=Y}</code>) (102_159)
<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=Decimal_Number}</code>) (460)
<code>\p{XPosixGraph}</code>	<code>\p{Graph}</code> (247_565)
<code>\p{XPosixLower}</code>	<code>\p{Lower}</code> (= <code>\p{Lowercase=Y}</code>) (/i= Cased=Yes) (1934)
<code>\p{XPosixPrint}</code>	<code>\p{Print}</code> (247_583)
<code>\p{XPosixPunct}</code>	<code>\p{Punct}</code> + ASCII-range <code>\p{Symbol}</code> (641)
<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) (1483)
<code>\p{XPosixWord}</code>	<code>\p{Word}</code> (103_406)
<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}</code>	<code>\p{Yijing_Hexagram_Symbols}</code> (= <code>\p{Block=Yijing_Hexagram_Symbols}</code>) (64)
X <code>\p{Yijing_Hexagram_Symbols}</code>	<code>\p{Block=Yijing_Hexagram_Symbols}</code> (Short: <code>\p{InYijing}</code>) (64)
<code>\p{Z}</code>	<code>\p{Separator}</code> (= <code>\p{General_Category=</code>

	Separator}) (20)
<code>\p{Zinh}</code>	<code>\p{Inherited}</code> (= <code>\p{Script=Inherited}</code>) (523)
<code>\p{Zl}</code>	<code>\p{Line_Separator}</code> (= <code>\p{General_Category=Line_Separator}</code>) (1)
<code>\p{Zp}</code>	<code>\p{Paragraph_Separator}</code> (= <code>\p{General_Category=Paragraph_Separator}</code>) (1)
<code>\p{Zs}</code>	<code>\p{Space_Separator}</code> (= <code>\p{General_Category=Space_Separator}</code>) (18)
<code>\p{Zyyy}</code>	<code>\p{Common}</code> (= <code>\p{Script=Common}</code>) (6413)
<code>\p{Zzzz}</code>	<code>\p{Unknown}</code> (= <code>\p{Script=Unknown}</code>) (1_003_930)
TX <code>\p{_\CanonDCIJ}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Soft_Dotted=Y}</code>) (46)
TX <code>\p{_\Case_Ignorable}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Case_Ignorable=Y}</code>) (1799)
TX <code>\p{_\CombAbove}</code>	(For internal use by Perl, not necessarily stable) (= <code>\p{Canonical_Combining_Class=Above}</code>) (349)

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 they exist 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{Canonical_Combining_Class=CCC133}
\p{Grapheme_Cluster_Break=Prepend}
\p{Joining_Type=Left_Joining}
```

Properties accessible through `Unicode::UCD`

All the Unicode character properties mentioned above (except for those marked as for internal use by Perl) are also accessible by `"prop_invlst()" in Unicode::UCD`.

Due to their nature, not all Unicode character properties are suitable for regular expression matches, nor `prop_invlst()`. The remaining non-provisional, non-internal ones are accessible via `"prop_invmmap()" in Unicode::UCD` (except for those that this Perl installation hasn't included; see *below for which those are*).

For compatibility with other parts of Perl, all the single forms given in the table in the *section above* are recognized. BUT, there are some ambiguities between some Perl extensions and the Unicode properties, all of which are silently resolved in favor of the official Unicode property. To avoid surprises, you should only use `prop_invmmap()` for forms listed in the table below, which omits the non-recommended ones. The affected forms are the Perl single form equivalents of Unicode properties, such as `\p{sc}` being a single-form equivalent of `\p{gc=sc}`, which is treated by `prop_invmmap()` as the `Script` property, whose short name is `sc`. The table indicates the current ambiguities in the INFO column, beginning with the word "NOT".

The standard Unicode properties listed below are documented in <http://www.unicode.org/reports/tr44/>; `Perl_Decimal_Digit` is documented in `"prop_invmmap()" in Unicode::UCD`. The other Perl extensions are in *"Other Properties" in perlunicode*;

The first column in the table is a name for the property; the second column is an alternative name, if any, plus possibly some annotations. The alternative name is the property's full name, unless that

would simply repeat the first column, in which case the second column indicates the property's short name (if different). The annotations are given only in the entry for the full name. If a property is obsolete, etc, the entry will be flagged with the same characters used in the table in the *section above*, like **D** or **S**.

NAME	INFO
Age	
AHex	ASCII_Hex_Digit
All	Any. (Perl extension)
Alnum	(Perl extension). Alphabetic and (decimal) Numeric
Alpha	Alphabetic
Alphabetic	(Short: Alpha)
Any	(Perl extension). [\x{0000}-\x{10FFFF}]
ASCII	Block=ASCII. (Perl extension). [[:ASCII:]]
ASCII_Hex_Digit	(Short: AHex)
Assigned	(Perl extension). All assigned code points
Bc	Bidi_Class
Bidi_C	Bidi_Control
Bidi_Class	(Short: bc)
Bidi_Control	(Short: Bidi_C)
Bidi_M	Bidi_Mirrored
Bidi_Mirrored	(Short: Bidi_M)
Bidi_Mirroring_Glyph	(Short: bmg)
Blank	(Perl extension). \h, Horizontal white space
Blk	Block
Block	(Short: blk)
Bmg	Bidi_Mirroring_Glyph
Canonical_Combining_Class	(Short: ccc)
Case_Folding	(Short: cf)
Case_Ignorable	(Short: CI)
Cased	
Category	General_Category
Ccc	Canonical_Combining_Class
CE	Composition_Exclusion
Cf	Case_Folding; NOT 'cf' meaning 'General_Category=Format'
Changes_When_Casefolded	(Short: CWCF)
Changes_When_Casemapped	(Short: CWCM)
Changes_When_Lowercased	(Short: CWL)
Changes_When_NFKC_Casefolded	(Short: CWKCF)
Changes_When_Titlecased	(Short: CWT)
Changes_When_Uppercased	(Short: CWU)
CI	Case_Ignorable
Cntrl	General_Category=Cntrl. (Perl extension). Control characters
Comp_Ex	Full_Composition_Exclusion
Composition_Exclusion	(Short: CE)
CWCF	Changes_When_Casefolded
CWCM	Changes_When_Casemapped
CWKCF	Changes_When_NFKC_Casefolded
CWL	Changes_When_Lowercased
CWT	Changes_When_Titlecased

CWU	Changes_When_Uppercased
Dash	
Decomposition_Mapping	(Short: dm)
Decomposition_Type	(Short: dt)
Default_Ignorable_Code_Point	(Short: DI)
Dep	Deprecated
Deprecated	(Short: Dep)
DI	Default_Ignorable_Code_Point
Dia	Diacritic
Diacritic	(Short: Dia)
Digit	General_Category=Digit. (Perl extension). [0-9] + all other decimal digits
Dm	Decomposition_Mapping
Dt	Decomposition_Type
Ea	East_Asian_Width
East_Asian_Width	(Short: ea)
Ext	Extender
Extender	(Short: Ext)
Full_Composition_Exclusion	(Short: Comp_Ex)
Gc	General_Category
GCB	Grapheme_Cluster_Break
General_Category	(Short: gc)
Gr_Base	Grapheme_Base
Gr_Ext	Grapheme_Extend
Graph	(Perl extension). Characters that are graphical
Grapheme_Base	(Short: Gr_Base)
Grapheme_Cluster_Break	(Short: GCB)
Grapheme_Extend	(Short: Gr_Ext)
Hangul_Syllable_Type	(Short: hst)
Hex	Hex_Digit
Hex_Digit	(Short: Hex)
HorizSpace	Blank. (Perl extension)
Hst	Hangul_Syllable_Type
D Hyphen	Supplanted by Line_Break property values; see www.unicode.org/reports/tr14
ID_Continue	(Short: IDC)
ID_Start	(Short: IDS)
IDC	ID_Continue
Ideo	Ideographic
Ideographic	(Short: Ideo)
IDS	ID_Start
IDS_Binary_Operator	(Short: IDSB)
IDS_Tertiary_Operator	(Short: IDST)
IDSB	IDS_Binary_Operator
IDST	IDS_Tertiary_Operator
In	Present_In. (Perl extension)
Isc	ISO_Comment; NOT 'isc' meaning 'General_Category=Other'
ISO_Comment	(Short: isc)
Jg	Joining_Group
Join_C	Join_Control
Join_Control	(Short: Join_C)
Joining_Group	(Short: jg)
Joining_Type	(Short: jt)
Jt	Joining_Type

Lb	Line_Break
Lc	Lowercase_Mapping; NOT 'lc' meaning 'General_Category=Cased_Letter'
Line_Break	(Short: lb)
LOE	Logical_Order_Exception
Logical_Order_Exception	(Short: LOE)
Lower	Lowercase
Lowercase	(Short: Lower)
Lowercase_Mapping	(Short: lc)
Math	
Na	Name
Na1	Unicode_1_Name
Name	(Short: na)
Name_Alias	
NChar	Noncharacter_Code_Point
NFC_QC	NFC_Quick_Check
NFC_Quick_Check	(Short: NFC_QC)
NFD_QC	NFD_Quick_Check
NFD_Quick_Check	(Short: NFD_QC)
NFKC_Casefold	(Short: NFKC_CF)
NFKC_CF	NFKC_Casefold
NFKC_QC	NFKC_Quick_Check
NFKC_Quick_Check	(Short: NFKC_QC)
NFKD_QC	NFKD_Quick_Check
NFKD_Quick_Check	(Short: NFKD_QC)
Noncharacter_Code_Point	(Short: NChar)
Nt	Numeric_Type
Numeric_Type	(Short: nt)
Numeric_Value	(Short: nv)
Nv	Numeric_Value
Pat_Syn	Pattern_Syntax
Pat_WS	Pattern_White_Space
Pattern_Syntax	(Short: Pat_Syn)
Pattern_White_Space	(Short: Pat_WS)
Perl_Decimal_Digit	(Perl extension)
PerlSpace	(Perl extension). \s, restricted to ASCII = [\f\n\r\t] plus vertical tab
PerlWord	(Perl extension). \w, restricted to ASCII = [A-Za-z0-9_]
PosixAlnum	(Perl extension). [A-Za-z0-9]
PosixAlpha	(Perl extension). [A-Za-z]
PosixBlank	(Perl extension). \t and ' '
PosixCntrl	(Perl extension). 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
PosixDigit	(Perl extension). [0-9]
PosixGraph	(Perl extension). [- !#\$%&'()*+,-./:;<>?@[\\]^_`{ }~0-9A-Za-z]
PosixLower	(Perl extension). [a-z]
PosixPrint	(Perl extension). [- 0-9A-Za- z!#\$%&'()*+,-./:;<>?@[\\]^_`{ }~]
PosixPunct	(Perl extension). [- !#\$%&'()*+,-./:;<>?@[\\]^_`{ }~]
PosixSpace	(Perl extension). \t, \n, \cK, \f, \r,

	and ' '. (\cK is vertical tab)
PosixUpper	(Perl extension). [A-Z]
PosixWord	PerlWord. (Perl extension)
PosixXDigit	(Perl extension). [0-9A-Fa-f]
Present_In	(Short: In). (Perl extension)
Print	(Perl extension). Characters that are graphical plus space characters (but no controls)
Punct	General_Category=Punct. (Perl extension)
QMark	Quotation_Mark
Quotation_Mark	(Short: QMark)
Radical	
SB	Sentence_Break
Sc	Script; NOT 'sc' meaning 'General_Category=Currency_Symbol'
Scf	Simple_Case_Folding
Script	(Short: sc)
Script_Extensions	(Short: scx)
Scx	Script_Extensions
SD	Soft_Dotted
Sentence_Break	(Short: SB)
Sfc	Simple_Case_Folding
Simple_Case_Folding	(Short: scf)
Simple_Lowercase_Mapping	(Short: slc)
Simple_Titlecase_Mapping	(Short: stc)
Simple_Uppercase_Mapping	(Short: suc)
Slc	Simple_Lowercase_Mapping
Soft_Dotted	(Short: SD)
Space	White_Space
SpacePerl	XPerlSpace. (Perl extension)
Stc	Simple_Titlecase_Mapping
STerm	
Suc	Simple_Uppercase_Mapping
Tc	Titlecase_Mapping
Term	Terminal_Punctuation
Terminal_Punctuation	(Short: Term)
Title	Titlecase. (Perl extension)
Titlecase	(Short: Title). (Perl extension). (= \p{Gc=Lt})
Titlecase_Mapping	(Short: tc)
Uc	Uppercase_Mapping
UIdeo	Unified_Ideograph
Unicode_1_Name	(Short: na1)
Unified_Ideograph	(Short: UIdeo)
Upper	Uppercase
Uppercase	(Short: Upper)
Uppercase_Mapping	(Short: uc)
Variation_Selector	(Short: VS)
VertSpace	(Perl extension). \v
VS	Variation_Selector
WB	Word_Break
White_Space	(Short: WSpace)
Word	(Perl extension). \w, including beyond ASCII; = \p{Alnum} + \pM + \p{Pc}
Word_Break	(Short: WB)
WSpace	White_Space

XDigit	(Perl extension)
XID_Continue	(Short: XIDC)
XID_Start	(Short: XIDS)
XIDC	XID_Continue
XIDS	XID_Start
XPerlSpace	(Perl extension). \s, including beyond ASCII
XPosixAlnum	Alnum. (Perl extension)
XPosixAlpha	Alpha. (Perl extension)
XPosixBlank	Blank. (Perl extension)
XPosixCntrl	General_Category=Cntrl. (Perl extension)
XPosixDigit	General_Category=Digit. (Perl extension)
XPosixGraph	Graph. (Perl extension)
XPosixLower	Lower. (Perl extension)
XPosixPrint	Print. (Perl extension)
XPosixPunct	(Perl extension). \p{Punct} + ASCII-range \p{Symbol}
XPosixSpace	Space. (Perl extension)
XPosixUpper	Upper. (Perl extension)
XPosixWord	Word. (Perl extension)
XPosixXDigit	XDigit. (Perl extension)

Properties accessible through other means

Certain properties are accessible also via core function calls. These are:

Lowercase_Mapping	lc() and lcfirst()
Titlecase_Mapping	ucfirst()
Uppercase_Mapping	uc()

Also, `Case_Folding` is accessible through the `/i` modifier in regular expressions, the `\F` transliteration escape, and the `fc` operator.

And, the `Name` and `Name_Aliases` properties are accessible through the `\N{ }` interpolation in double-quoted strings and regular expressions; and functions `charnames::viacode()`, `charnames::vianame()`, and `charnames::string_vianame()` (which require a `use charnames ();` to be specified).

Finally, most properties related to decomposition are accessible via `Unicode::Normalize`.

Unicode character properties that are NOT accepted by Perl

Perl will generate an error for a few character properties in Unicode when used in a regular expression. The non-Unihan ones are listed below, with the reasons they are not accepted, perhaps with work-arounds. The short names for the properties are listed enclosed in (parentheses). As described after the list, an installation can change the defaults and choose to accept any of these. The list is machine generated based on the choices made for the installation that generated this document.

`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)

Indic_Matra_Category (InMC)

Indic_Syllabic_Category (InSC)

Provisional

Jamo_Short_Name (JSN)

Other_Alphabetic (OAlpha)

Other_Default_Ignorable_Code_Point (ODI)

Other_Grapheme_Extend (OGr_Ext)

Other_ID_Continue (OIDC)

Other_ID_Start (OIDS)

Other_Lowercase (OLower)

Other_Math (OMath)

Other_Uppercase (OUpper)

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". Consider instead using "Script_Extensions=Katakana" or "Script_Extensions=Hiragana" (or both)

Script_Extensions=Katakana_Or_Hiragana (scx=Hrkt)

All code points that would be matched by this are matched by either "Script_Extensions=Katakana" or "Script_Extensions=Hiragana"

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).

Other information in the Unicode data base

The Unicode data base is delivered in two different formats. The XML version is valid for more modern Unicode releases. The other version is a collection of files. The two are intended to give equivalent information. Perl uses the older form; this allows you to recompile Perl to use early Unicode releases.

The only non-character property that Perl currently supports is Named Sequences, in which a sequence of code points is given a name and generally treated as a single entity. (Perl supports these via the `\N{...}` double-quotish construct, "`chardata::string_via_name(name)`" in `chardata`, and "`namedseq()`" in `Unicode::UCD`).

Below is a list of the files in the Unicode data base that Perl doesn't currently use, along with very brief descriptions of their purposes. Some of the names of the files have been shortened from those that Unicode uses, in order to allow them to be distinguishable from similarly named files on file systems for which only the first 8 characters of a name are significant.

auxiliary/GraphemeBreakTest.html

auxiliary/LineBreakTest.html

auxiliary/SentenceBreakTest.html

auxiliary/WordBreakTest.html

Documentation of validation tests

auxiliary/LBTest.txt

auxiliary/SBTest.txt

auxiliary/WBTest.txt

BidiTest.txt

NormTest.txt

Validation Tests

CJKRadicals.txt

Maps the kRSUnicode property values to corresponding code points

EmojiSources.txt

Maps certain Unicode code points to their legacy Japanese cell-phone values

Index.txt

Alphabetical index of Unicode characters

IndicMatraCategory.txt

IndicSyllabicCategory.txt

Provisional; for the analysis and processing of Indic scripts

NamedSqProv.txt

Named sequences proposed for inclusion in a later version of the Unicode Standard; if you need them now, you can append this file to *NamedSequences.txt* and recompile perl

NamesList.txt

Annotated list of characters

NormalizationCorrections.txt

Documentation of corrections already incorporated into the Unicode data base

Props.txt

Only in very early releases; is a subset of *PropList.txt* (which is used instead)

ReadMe.txt

Documentation

StandardizedVariants.txt

Certain glyph variations for character display are standardized. This lists the non-Unihan ones; the Unihan ones are also not used by Perl, and are in a separate Unicode data base

<http://www.unicode.org/ivd>

USourceData.pdf

USourceData.txt

Documentation of status and cross reference of proposals for encoding by Unicode of Unihan characters

SEE ALSO

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

perlrecharclass

perlunicode