This page lists the regular expression syntax accepted by RE2. It also lists syntax accepted by PCRE, PERL, and VIM. Grayed out expressions are not supported by RE2.

**Single characters:**
- .  
  any character, possibly including newline (s=true)
- [xyz]  
  character class
- [^xyz]  
  negated character class
- \d  
  Perl character class
- \D  
  negated Perl character class
- [\:alpha:]  
  ASCII character class
- [^[\:alpha:]  
  negated ASCII character class
- \p{N}  
  Unicode character class (one-letter name)
- \p{Greek}  
  Unicode character class
- \PN  
  negated Unicode character class (one-letter name)
- \P{Greek}  
  negated Unicode character class

**Composites:**
- xy  
  x followed by y
- x|y  
  x or y (prefer x)

**Repetitions:**
- x*  
  zero or more x, prefer more
- x+  
  one or more x, prefer more
- x?  
  zero or one x, prefer one
- x{n,m}  
  n or n+1 or ... or m x, prefer more
- x(n{)}  
  n or more x, prefer more
- x(n)  
  exactly n x
- x{n,}  
  zero or more x, prefer fewer
- x{n,}+  
  one or more x, prefer fewer
- x{n,}?  
  zero or one x, prefer zero
- x{n,m}?  
  n or n+1 or ... or m x, prefer fewer
- x{n,}?  
  n or more x, prefer fewer
- x{n}+  
  exactly n x
- a{n,}+  
  n or more x, possessive (NOT SUPPORTED)
- x{n}+  
  one or more x, possessive (NOT SUPPORTED)
- x{n,}+  
  zero or one x, possessive (NOT SUPPORTED)
- x{n,m}+  
  n or ... or m x, possessive (NOT SUPPORTED)
- x{n,}+  
  n or more x, possessive (NOT SUPPORTED)
- x{n}+  
  exactly n x, possessive (NOT SUPPORTED)

**Grouping:**
- (re)  
  numbered capturing group
- (?!<name>re)  
  named & numbered capturing group
- (?<name>re)  
  named & numbered capturing group (NOT SUPPORTED)
- (?<!name>re)  
  named & numbered capturing group (NOT SUPPORTED)
- (?<re)  
  non-capturing group
- (?flags)  
  set flags within current group; non-capturing
(?!flags:re) set flags during re; non-capturing
(?!text) comment (NOT SUPPORTED)
(?!x|y|z) branch numbering reset (NOT SUPPORTED)
(?!>re) possessive match of re (NOT SUPPORTED) VIM
(?re>) possessive match of re (NOT SUPPORTED) VIM
\%(re) non-capturing group (NOT SUPPORTED) VIM

Flags:

i case-insensitive (default false)
m multi-line mode: ^ and $ match begin/end line in addition to begin/end text (default false)
s let. match \n (default false)
U ungreedy: swap meaning of x* and x*?, x+ and x+?, etc (default false)

Flag syntax is xyz (set) or -xyz (clear) or xy-z (set xy, clear z).

Empty strings:

^ at beginning of text or line (m=true)
$ at end of text (like \z not \Z)
\A at beginning of text
\b at word boundary (\w on one side and \W, \A, or \z on the other)
\G at beginning of subtext being searched (NOT SUPPORTED) PCRE
\G at end of last match (NOT SUPPORTED) PERL
\Z at end of text, or before newline at end of text (NOT SUPPORTED)
\v at end of text
(?!re) before text matching re (NOT SUPPORTED)
(?!re) before text not matching re (NOT SUPPORTED)
(<=re) after text matching re (NOT SUPPORTED)
(<re) after text matching re (NOT SUPPORTED) VIM
@re before text matching re (NOT SUPPORTED) VIM
@re before text not matching re (NOT SUPPORTED) VIM
@<=re after text matching re (NOT SUPPORTED) VIM
@<re after text not matching re (NOT SUPPORTED) VIM
\zs sets start of match (= \K) (NOT SUPPORTED) VIM
\ze sets end of match (NOT SUPPORTED) VIM
\%^ beginning of file (NOT SUPPORTED) VIM
\%$ end of file (NOT SUPPORTED) VIM
\%V on screen (NOT SUPPORTED) VIM
\%# cursor position (NOT SUPPORTED) VIM
\%m mark m position (NOT SUPPORTED) VIM
\%23| in line 23 (NOT SUPPORTED) VIM
\%23c in column 23 (NOT SUPPORTED) VIM
\%23v in virtual column 23 (NOT SUPPORTED) VIM

Escape sequences:

\a bell (≡ \007)
\f form feed (≡ \014)
\t horizontal tab (≡ \011)
\n newline (≡ \012)
\r carriage return (≡ \015)
\v vertical tab character (≡ \013)
* literal *, for any punctuation character *
123 octal character code (up to three digits)
\7hex character code (exactly two digits)
\x{10FFFF} hex character code
\C match a single byte even in UTF-8 mode
\Q...\E literal text ... even if ... has punctuation
\1 backreference (NOT SUPPORTED)
\b backspace (NOT SUPPORTED) (use \010)
\cK control char *K (NOT SUPPORTED) (use \001 etc)
\e escape (NOT SUPPORTED) (use \033)
\g1 backreference (NOT SUPPORTED)
\g{1} backreference (NOT SUPPORTED)
\g{+1} backreference (NOT SUPPORTED)
Character class elements:
  x           single character
  A-Z         character range (inclusive)
  \d          Perl character class
  [:foo:]     ASCII character class foo
  \p{Foo}     Unicode character class Foo
  \pF         Unicode character class F (one-letter name)

Named character classes as character class elements:
  [\d]        digits (≡ [0-9])
  [^\d]       not digits (≡ [^0-9])
  [D]         not digits (≡ [0-9])
  [^D]        not not digits (≡ [^0-9])
  [:name:]    named ASCII class inside character class (≡ [:name:])
  [^:name:]   named ASCII class inside negated character class (≡ [^:name:])
  \p{Name}   named Unicode property inside character class (≡ \p{Name})
  [^\p{Name}]  named Unicode property inside negated character class (≡ \P{Name})

Perl character classes:
  \d          digits (≡ [0-9])
  \D          not digits (≡ [^0-9])
  \s          whitespace (≡ [\t\n\f\r \ ])
  \S          not whitespace (≡ [^\t\n\f\r \ ])
  \w          word characters (≡ [0-9A-Za-z\_])
  \W          not word characters (≡ [^0-9A-Za-z\_])
  \h          horizontal space (NOT SUPPORTED)
  \H          not horizontal space (NOT SUPPORTED)
  \v          vertical space (NOT SUPPORTED)
  \V          not vertical space (NOT SUPPORTED)

ASCII character classes:
  [:alnum:]    alphanumeric (≡ [0-9A-Za-z])
  [:alpha:]    alphabetic (≡ [A-Za-z])
  [:ascii:]    ASCII (≡ [\x00-\x7F])
  [:blank:]    blank (≡ [ \t])
  [:cntrl:]    control (≡ [\x00-\x1F\x7F])
  [:digit:]    digits (≡ [0-9])
  [:graph:]    graphical (≡ [+-*/\^\%&\()\+=<=>?@\{\}\[\]\_\{\}~\])
  [:lower:]    lower case (≡ [a-z])
  [:print:]    printable (≡ [\-\+\-\%\&\()\+=<=>?@\{\}\[\]\_\{\}~\])
  [:punct:]    punctuation (≡ [\-\%\&\()\+=<=>?@\{\}\[\]\_\{\}~\])
  [:space:]    whitespace (≡ [\t\n\f\r \ ])
  [:upper:]    upper case (≡ [A-Z])
  [:word:]     word characters (≡ [0-9A-Za-z\_])
Unicode character class names—general category:

- C: other
- Cc: control
- Cf: format
- Cn: unassigned code points (NOT SUPPORTED)
- Co: private use
- Cs: surrogate
- L: letter
- LC: cased letter (NOT SUPPORTED)
- Lc: cased letter (NOT SUPPORTED)
- Li: lowercase letter
- Lm: modifier letter
- Lo: other letter
- Lt: titlecase letter
- Lu: uppercase letter
- M: mark
- Mc: spacing mark
- Me: enclosing mark
- Mn: non-spacing mark
- N: number
- Nd: decimal number
- Ni: letter number
- No: other number
- P: punctuation
- Pc: connector punctuation
- Pd: dash punctuation
- Pe: close punctuation
- Pf: final punctuation
- Pi: initial punctuation
- Po: other punctuation
- Ps: open punctuation
- S: symbol
- Sc: currency symbol
- Sk: modifier symbol
- Sm: math symbol
- So: other symbol
- Z: separator
- Zl: line separator
- Zp: paragraph separator
- Zs: space separator

Unicode character class names—scripts:

- Arabic: Arabic
- Armenian: Armenian
- Balinese: Balinese
- Bamum: Bamum
- Batak: Batak
- Bengali: Bengali
- Bopomofo: Bopomofo
- Brahmi: Brahmi
- Braille: Braille
- Buginese: Buginese
- Buhid: Buhid
- Canadian_Aboriginal: Canadian Aboriginal
- Carian: Carian
- Chakma: Chakma
- Cham: Cham
- Cherokee: Cherokee
- Common: characters not specific to one script
- Coptic: Coptic
- Cuneiform: Cuneiform
<table>
<thead>
<tr>
<th>Script</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cypriot</td>
<td>Cypriot</td>
</tr>
<tr>
<td>Cyrillic</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>Deseret</td>
<td>Deseret</td>
</tr>
<tr>
<td>Devanagari</td>
<td>Devanagari</td>
</tr>
<tr>
<td>Egyptian_Hieroglyphs</td>
<td>Egyptian Hieroglyphs</td>
</tr>
<tr>
<td>Ethiopic</td>
<td>Ethiopic</td>
</tr>
<tr>
<td>Georgian</td>
<td>Georgian</td>
</tr>
<tr>
<td>Glagolitic</td>
<td>Glagolitic</td>
</tr>
<tr>
<td>Gothic</td>
<td>Gothic</td>
</tr>
<tr>
<td>Greek</td>
<td>Greek</td>
</tr>
<tr>
<td>Gujarati</td>
<td>Gujarati</td>
</tr>
<tr>
<td>Gurmukhi</td>
<td>Gurmukhi</td>
</tr>
<tr>
<td>Han</td>
<td>Han</td>
</tr>
<tr>
<td>Hangul</td>
<td>Hangul</td>
</tr>
<tr>
<td>Hanunoo</td>
<td>Hanunoo</td>
</tr>
<tr>
<td>Hebrew</td>
<td>Hebrew</td>
</tr>
<tr>
<td>Hiragana</td>
<td>Hiragana</td>
</tr>
<tr>
<td>Imperial_Aramaic</td>
<td>Imperial Aramaic</td>
</tr>
<tr>
<td>Inherited</td>
<td>Inherit script from previous character</td>
</tr>
<tr>
<td>Inscriptional_Pahlavi</td>
<td>Inscriptional Pahlavi</td>
</tr>
<tr>
<td>Inscriptional_Partihan</td>
<td>Inscriptional Partihan</td>
</tr>
<tr>
<td>Javanese</td>
<td>Javanese</td>
</tr>
<tr>
<td>Kairhi</td>
<td>Kairhi</td>
</tr>
<tr>
<td>Kannada</td>
<td>Kannada</td>
</tr>
<tr>
<td>Katakana</td>
<td>Katakana</td>
</tr>
<tr>
<td>Kayah_Li</td>
<td>Kayah Li</td>
</tr>
<tr>
<td>Kharoshthi</td>
<td>Kharoshthi</td>
</tr>
<tr>
<td>Khmer</td>
<td>Khmer</td>
</tr>
<tr>
<td>Lao</td>
<td>Lao</td>
</tr>
<tr>
<td>Latin</td>
<td>Latin</td>
</tr>
<tr>
<td>Lepcha</td>
<td>Lepcha</td>
</tr>
<tr>
<td>Limbu</td>
<td>Limbu</td>
</tr>
<tr>
<td>Linear_B</td>
<td>Linear B</td>
</tr>
<tr>
<td>Lyceian</td>
<td>Lyceian</td>
</tr>
<tr>
<td>Lydian</td>
<td>Lydian</td>
</tr>
<tr>
<td>Malayalam</td>
<td>Malayalam</td>
</tr>
<tr>
<td>Mandaic</td>
<td>Mandaic</td>
</tr>
<tr>
<td>Meetei_Mayek</td>
<td>Meetei Mayek</td>
</tr>
<tr>
<td>Meroitic_Cursive</td>
<td>Meroitic Cursive</td>
</tr>
<tr>
<td>Meroitic_Hieroglyphs</td>
<td>Meroitic Hieroglyphs</td>
</tr>
<tr>
<td>Miao</td>
<td>Miao</td>
</tr>
<tr>
<td>Mongolian</td>
<td>Mongolian</td>
</tr>
<tr>
<td>Myanmar</td>
<td>Myanmar</td>
</tr>
<tr>
<td>New_Tai_Lue</td>
<td>New Tai Lue (aka Simplified Tai Lue)</td>
</tr>
<tr>
<td>Nko</td>
<td>Nko</td>
</tr>
<tr>
<td>Ogham</td>
<td>Ogham</td>
</tr>
<tr>
<td>Ol_Chiki</td>
<td>Ol Chiki</td>
</tr>
<tr>
<td>Old_Italic</td>
<td>Old Italic</td>
</tr>
<tr>
<td>Old_Persian</td>
<td>Old Persian</td>
</tr>
<tr>
<td>Old_South_Arabian</td>
<td>Old South Arabian</td>
</tr>
<tr>
<td>Old_Turkic</td>
<td>Old Turkic</td>
</tr>
<tr>
<td>Oriya</td>
<td>Oriya</td>
</tr>
<tr>
<td>Osmanya</td>
<td>Osmanya</td>
</tr>
<tr>
<td>Phags_Pa</td>
<td>Phags Pa</td>
</tr>
<tr>
<td>Phoenician</td>
<td>Phoenician</td>
</tr>
<tr>
<td>Rejang</td>
<td>Rejang</td>
</tr>
<tr>
<td>Runic</td>
<td>Runic</td>
</tr>
<tr>
<td>Saurashtra</td>
<td>Saurashtra</td>
</tr>
<tr>
<td>Sharada</td>
<td>Sharada</td>
</tr>
<tr>
<td>Shavian</td>
<td>Shavian</td>
</tr>
<tr>
<td>Sinhala</td>
<td>Sinhala</td>
</tr>
<tr>
<td>Sora_Sompeng</td>
<td>Sora Sompeng</td>
</tr>
</tbody>
</table>

*Syntax - re2 - I define UNIX as "30 definitions of regular expressions living under one roof." —Don Knuth - an efficient, principled regular expression library - Google Project Hosting*
<table>
<thead>
<tr>
<th>Language</th>
<th>Vim character classes</th>
<th>Vim flags</th>
<th>Magic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sundanese</td>
<td>identifier character (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syloti_Nagri</td>
<td>\i except digits (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syriac</td>
<td>\k keyword character (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tagalog</td>
<td>\k except digits (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tagbanwa</td>
<td>\f file name character (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tai_Le</td>
<td>\F except digits (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tai_Tham</td>
<td>\s whitespace character (≡ [ \t]) (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tai_Viet</td>
<td>\x hex digits (≡ [0-9A-Fa-f]) (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telugu</td>
<td>\o octal digits (≡ [0-7]) (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thaana</td>
<td>\p printable character (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thai</td>
<td>\D not \d VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tibetan</td>
<td>\x plus newline, for any \x (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tifinagh</td>
<td>\l lowercase (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ugartic</td>
<td>\a alphabetic (NOT SUPPORTED) VIM</td>
<td></td>
<td>LET</td>
</tr>
<tr>
<td>Vai</td>
<td>\A not \a (NOT SUPPORTED) VIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yi</td>
<td>\L lowercase (NOT SUPPORTED) VIM</td>
<td></td>
<td>LET</td>
</tr>
<tr>
<td>Vim character classes:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vim flags:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magic:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax - re2** - I define UNIX as "30 definitions of regular expressions living under one roof." —Don Knuth - an efficient, principled regular expression library - Google Project Hosting

02/07/2014

https://code.google.com/p/re2/wiki/Syntax
WTF?

So the major trade offs are back-references for example: (abc)\1 and not matching look-behinds. In exchange you get high speed regex. There are wrappers for many popular languages too. This can be very useful for certain applications.

For definitions of various unicode classes for \p{Class} syntax, the following data files are useful:

- http://unicode.org/Public/UNIDATA/PropertyValueAliases.txt
- http://www.unicode.org/Public/UNIDATA/UnicodeData.txt

(*UCP) is missing from this list. In PCRE, this sets the PCRE_UCP option, which adds unicode property support, and changes \w to match alphanumeric from unicode properties.

If I try \d{3} for entering three digits the expression does not match. I guess I am reading this wrongly...

ok, it's a good way to learn golang

jvar..., try using \d{3}.

"&lt;img src=x onerror=confirm(1);&gt;"
Comment by d...@google.com, Mar 21, 2014

"Grayed out expressions are not supported by RE2" would be fine if the gray could be readily discerned from black. Please use a color (like red) or italic or strikeout, or actually use a light gray background (or even better, a combination).

Comment by awbloc...@google.com, Apr 29, 2014

+1 for strikeout.

Comment by rghave...@gmail.com, Jun 18, 2014

+1 for color or strikeout, its painful to differentiate between supported and non-supported.

Comment by mbotic1...@googlemail.com, Jun 19, 2014

+1 for color (#ff0000)

Comment by sarun...@gmail.com, Jun 27 (4 days ago)

For those who want to change color of (NOT SUPPORTED) items in this page. Just enter this into developer console:

```javascript
(function(){
 var allTr=document.getElementsByTagName("tr");
 for(var i=0; i<allTr.length; i++)
 if((allTr[i].innerHTML.indexOf("NOT SUPPORT")>=0) &
 !(allTr[i].innerHTML.indexOf("<tr">=0)))(
 allTr[i].style.color="red";
 }
}());
```

[Sign in](https://code.google.com/p/re2/wiki/Syntax) to add a comment