

Table of Contents

Regular Expressions. 2

Regular Expressions.

Summary of regular-expression constructs

Construct	Matches
Characters	
X	The character x
\\	The backslash character
\0n	The character with octal value 0n (0 <= n <= 7)
\0nn	The character with octal value 0nn (0 <= n <= 7))
\0mnn	The character with octal value 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	The character with hexadecimal value 0xhh
\uhhhh	The character with hexadecimal value 0xhhhh
\t	The tab character ('\u0009')
\n	The newline (line feed) character ('\u000A')
\r	The carriage-return character ('\u000D')
\f	The form-feed character ('\u000C')
\a	The alert (bell) character ('\u0007')
\e	The escape character ('\u001B')
\cx	The control character corresponding to x
Character classes	
[abc]	a, b, or c (simple class)
[abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)
[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)
[a-z&&[def]]	d, e, or f (intersection)
[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)
[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z](subtraction)
Predefined character classes	
.	Any character (may or may not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]
POSIX character classes (US-ASCII only)	
\p{Lower}	A lower-case alphabetic character: [a-z]
\p{Upper}	An upper-case alphabetic character:[A-Z]
\p{ASCII}	All ASCII:[\x00-\x7F]
\p{Alpha}	An alphabetic character:[\p{Lower}\p{Upper}]
\p{Digit}	A decimal digit: [0-9]
\p{Alnum}	An alphanumeric character:[\p{Alpha}\p{Digit}]
\p{Punct}	Punctuation: One of !"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~
\p{Graph}	A visible character: [\p{Alnum}\p{Punct}]

Construct	Matches
Characters	
<code>\p{Print}</code>	A printable character: <code>[\p{Graph}]</code>
<code>\p{Blank}</code>	A space or a tab: <code>[\ \t]</code>
<code>\p{Cntrl}</code>	A control character: <code>[\x00-\x1F\x7F]</code>
<code>\p{XDigit}</code>	A hexadecimal digit: <code>[0-9a-fA-F]</code>
<code>\p{Space}</code>	A whitespace character: <code>[\ \t\n\x0B\f\r]</code>
Classes for Unicode blocks and categories	
<code>\p{InGreek}</code>	A character in the Greek block (simple block)
<code>\p{Lu}</code>	An uppercase letter (simple category)
<code>\p{Sc}</code>	A currency symbol
<code>\P{InGreek}</code>	Any character except one in the Greek block (negation)
<code>[\p{L}&&[^\p{Lu}]]</code>	Any letter except an uppercase letter (subtraction)
Boundary matchers	
<code>^</code>	The beginning of a line
<code>\$</code>	The end of a line
<code>\b</code>	A word boundary
<code>\B</code>	A non-word boundary
<code>\A</code>	The beginning of the input
<code>\G</code>	The end of the previous match
<code>\Z</code>	The end of the input but for the final terminator , if any
<code>\z</code>	The end of the input
Greedy quantifiers	
<code>X?</code>	X, once or not at all
<code>X*</code>	X, zero or more times
<code>X+</code>	X, one or more times
<code>X{n}</code>	X, exactly n times
<code>X{n,}</code>	X, at least n times
<code>X{n,m}</code>	X, at least n but not more than m times
Reluctant quantifiers	
<code>X??</code>	X, once or not at all
<code>X*?</code>	X, zero or more times
<code>X+?</code>	X, one or more times
<code>X{n}?</code>	X, exactly n times
<code>X{n,}?</code>	X, at least n times
<code>X{n,m}?</code>	X, at least n but not more than m times
Possessive quantifiers	
<code>X?+</code>	X, once or not at all
<code>X*+</code>	X, zero or more times
<code>X++</code>	X, one or more times
<code>X{n}+</code>	X, exactly n times
<code>X{n,}+</code>	X, at least n times
<code>X{n,m}+</code>	X, at least n but not more than m times
Logical operators	
<code>XY</code>	X followed by Y
<code>X Y</code>	Either X or Y

Construct	Matches
Characters	
(X)	X, as a capturing group
Back references	
\n	Whatever the n th capturing group matched
Quotation	
\	Nothing, but quotes the following character
\Q	Nothing, but quotes all characters until \E
\E	Nothing, but ends quoting started by \Q
Special constructs (non-capturing)	
(?:X)	X, as a non-capturing group
(?idmsux-idmsux)	Nothing, but turns match flags on - off
(?idmsux-idmsux:X)	X, as a non-capturing group with the given flags on - off
(?=X)	X, via zero-width positive lookahead
(?!X)	X, via zero-width negative lookahead
(?<=X)	X, via zero-width positive lookbehind
(?<!X)	X, via zero-width negative lookbehind
(?>X)	X, as an independent, non-capturing group

From:

<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:

http://www.awareim.com/dokuwiki/docs/8000_appendices/0500_regular_expressions

Last update: **2026/01/02 10:02**

