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
\x{h...h}	The character with hexadecimal value 0xh...h (Character.MIN_CODE_POINT <= 0xh...h <= Character.MAX_CODE_POINT)
\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]
\h	A horizontal whitespace character: [\t\xA0\u1680\u180e\u2000-\u200a\u202f\u205f\u3000]
\H	A non-horizontal whitespace character: [^\h]
\s	A whitespace character: [\t\n\x0B\f\r]

\S	A non-whitespace character: [^\s]
\v	A vertical whitespace character: [\n\x0B\f\r\x85\u2028\u2029]
\V	A non-vertical whitespace character: [^\v]
\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}]
\p{Print}	A printable character: [\p{Graph}\x20]
\p{Blank}	A space or a tab: [\t]
\p{Cntrl}	A control character: [\x00-\x1F\x7F]
\p{XDigit}	A hexadecimal digit: [0-9a-fA-F]
\p{Space}	A whitespace character: [\t\n\x0B\f\r]

java.lang.Character classes (simple java character type)

\p{javaLowerCase}	Equivalent to java.lang.Character.isLowerCase()
\p{javaUpperCase}	Equivalent to java.lang.Character.isUpperCase()
\p{javaWhitespace}	Equivalent to java.lang.Character.isWhitespace()
\p{javaMirrored}	Equivalent to java.lang.Character.isMirrored()

Classes for Unicode scripts, blocks, categories and binary properties
----

\p{IsLatin}	A Latin script character (script)
\p{InGreek}	A character in the Greek block (block)
\p{Lu}	An uppercase letter (category)
\p{IsAlphabetic}	An alphabetic character (binary property)
\p{Sc}	A currency symbol
\P{InGreek}	Any character except one in the Greek block (negation)
[\p{L}&&[^\p{Lu}]]	Any letter except an uppercase letter (subtraction)

Boundary matchers
----

^	The beginning of a line
$	The end of a line

\b	A word boundary
\B	A non-word boundary
\A	The beginning of the input
\G	The end of the previous match
\Z	The end of the input but for the final terminator, if any
\z	The end of the input

Linebreak matcher

\R	Any Unicode linebreak sequence, is equivalent to \u000D\u000A|[\u000A\u000B\u000C\u000D\u0085\u2028\u2029]

Greedy quantifiers

X?	X, once or not at all
X*	X, zero or more times
X+	X, one or more times
X{n}	X, exactly n times
X{n,}	X, at least n times
X{n,m}	X, at least n but not more than m times

Reluctant quantifiers

X??	X, once or not at all
X*?	X, zero or more times
X+?	X, one or more times
X{n}?	X, exactly n times
X{n,}?	X, at least n times
X{n,m}?	X, at least n but not more than m times

Possessive quantifiers

X?+	X, once or not at all
X*+	X, zero or more times
X++	X, one or more times
X{n}+	X, exactly n times
X{n,}+	X, at least n times
X{n,m}+	X, at least n but not more than m times

Logical operators

XY	X followed by Y
X|Y	Either X or Y

(X)	X, as a capturing group

Back references

\n	Whatever the nth capturing group matched
\k<name>	Whatever the named-capturing group "name" 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 (named-capturing and non-capturing)

(?<name>X)	X, as a named-capturing group
(?:X)	X, as a non-capturing group
(?idmsuxU-idmsuxU)	Nothing, but turns match flags i d m s u x U on - off
(?idmsux-idmsux:X)	X, as a non-capturing group with the given flags i d m s u x 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