RegEx Cheat Sheet

Common Tokens

A single character of: a, b, or c	[abc]	Capture enclosed	(...)
A character except: a, b, or c	[^abc]	Match either a or b	(a|b)
A character in the range: a-z	[a-z]	Zero or one of a	a?
A chapter not in range: a-z	[^a-z]	Zero of more of a	a*
A character in the range: a-z or A-Z	[a-zA-Z]	One or more of a	a+
Any single character	.	Exactly 3 of a	a{3}
Any Whitespace character	\s	3 or more of a	a{3,}
Any non-whitespace character	\S	Between 3 and 6 of a	a{3,6}
Any digit	\d	Start of string	^
Any non-digit	\D	End of string	$
Any word character	\w	A word boundary	\n
Any non-word character	\W	None-word boundary	\B

Meta Sequence

Any Unicode sequences, line breaks included	\X
Match one data unit	\C
Unicode newlines	\R
Vertical whitespace character	\v
Negation of \v	\V
Horizontal whitespace character	\h
Negation of	\H
Reset match	\K
Match nth sub pattern	\n
Unicode propert X	\pX
Unicode property or script category	\p{...}
Negation of \pX	\PX
Negation of \p	\P{...}
Quote; treat as literals	\Q...\E
Match sub pattern `name`	\k<name>
Match sub pattern `name`	\k'name'
Match sub pattern `name`	\k{name}
Match nth sub pattern	\gn
Match nth sub pattern	\g{n}
Recurse nth capture group	\g<n>
Recurses nth capture group	\g'n'
Match nth relative previous sub pattern	\g{-n}
Recurse nth relative upcoming sub pattern	\g<+n>
Match nth relative upcoming submitter	\g'+n'
Recurse names capture group	`letter`
Match previously-names capture group `letter`	\g{letter}
Recurses names capture group `letter’	\g<letter>
Hex character YY	\xYY
Hex characer YYYY	\x{YYYY}
Octal character ddd	\ddd
Control character Y	\cY
Backspace character	[\b]
Makes any character literal	\

General Tokens

Newline	\n
Carriage return	\r
Tab	\t
Null character	\0

Character Class Modifiers

A single character of: a, b, or c	[abc]
A character except: a, b, or c	[^abc]
A character in the range: a-z	[a-z]
A chapter not in range: a-z	[^a-z]
A character in the range: a-z or A-Z	[a-zA-Z]
Letters and digits	[[:alnum:]]
Letters	[[:alpha:]]
ASCII codes 0-127	[[:ascii:]]
Space or tab only	[[:blank:]]
Control characters	[[:cntrl:]]
Decimal digits	[[:digit:]]
Visible characters (not space)	[[:graph:]]
Lowercase letters	[[:lower:]]
Uppercase letter	[[:upper:]]
Word characters	[[:word:]]
	[[:xdigit:]]
	[[:<:]]
	[[:>:]]

braze

RegEx Cheat Sheet

Group Constants

Capture everything enclosed	(...)
Match either a or b	(a|b)
Match everything enclosed	(?:...)
Atomic grouping (non-capturing)	(?>...)
Duplicate sub pattern group number	(?|...)
Comment	(?#...)
Named Capturing Group	(?'name'...)
Named Capturing Group	(?<name>...)
Named Capturing Group	(?P<name>...)
Inline modifiers	(?imsxXU)
Conditional statements	(?(1)yes|no)
Recursive Conditional Statement	(?(R#)yes|no)
Conditional statement	(?(R&name)yes|no)
Lookahead conditional	(?(?=...)yes|no)
Lookbehind conditional	(?(?<=...)yes|no)
Recurse entire pattern	(?R)
Recurse first sub pattern	(?1)
Recurse first relative sub pattern	(?+1)
Recurse subpattern `name`	(?&name)
Match subpattern `name`	(?P=name)
Recurse subpattern `name`	(?P>name)
Pre-define patterns before use	(?(DEFINE)...)
Positive Lookahead	(?=...)
Negative Lookahead	(?!...)
Positive Lookbehind	(?<=...)
Negative Lookbehind	(?<!...)
Control verb	(*ACCEPT)
Control verb	(*FAIL)
Control verb	(*MARK:NAME)
Control verb	(*COMMIT)
Control verb	(*PRUNE)
Control verb	(*SKIP)
Control verb	(*THEN)
Pattern modifier	(*UTF)
Pattern modifier	(*UTF8)
Pattern modifier	(*UTF16)
Pattern modifier	(*UTF32)
Pattern modifier	(*UCP)
Line break modifier	(*CR)
Line break modifier	(*LF)
Line break modifier	(*CRLF)
Line break modifier	(*ANYCRLF)
Line break modifier	(*ANY)
Line break modifier	\R
Line break modifier	(*BSR_ANYCRLF)
Line break modifier	(*BSR_UNICODE)
Regex engine modifier	(*LIMIT_MATCH=x)
Regex engine modifier	(*LIMIT_RECURSION=d)
Regex engine modifier	(*NO_AUTO_POSSESS)
Regex engine modifier	(*NO_START_OPT)

Quantifiers

Zero or one of a	a?
Zero of more of a	a*
One or more of a	a+
Exactly 3 of a	a{3}
3 or more of a	a{3,}
Between 3 and 6 of a	a{3,6}
Greedy quantifier	a*
Lazy quantifier	a*?
Possessive quantifier	a*+

Anchors

Start of match	\G
Start of string	^
End of string	$
Start of string	\A
End of string	\Z
Absolute end of string	\z
A word boundary	\b
A non-word boundary	\B

Flags/Modifiers

Global	g
Multiline	m
Case sensitive	l
Ignore whitespace	x
Single line	s
Unicode	u
eXtended	X
Ungreedy	U
Anchor	A
Duplicate group names	J

Substitution

Complete match contents	\0
Contents in capture group 1	\1 or $1
Contents in capture group `foo`	${foo}
Hexidecimal replacement values	\x20, \x{06fa}
Tab	\t
Carriage return	\r
Newline	\n
Form-feed	\f
Uppercase Transformation	\U
Lowercase transformation	\L
Terminate any Transformation	\E