vitorbritto / **regex.md**

Created 4 years ago • Report gist

Regex Cheat Sheet

`regex.md`

Regular Expressions

Basic Syntax

- `/.../` : Start and end regex delimiters
- `|` : Alternation
- `()` : Grouping

Position Matching

- `^` : Start of string or start of line in multi-line mode
- `\A` : Start of string
- `$` : End of string or end of line in multi-line mode
- `\Z` : End of string
- `\b` : Word boundary
- `\B` : Not word boundary
- `\<` : Start of word
- `\>` : End of word

Character Classes

- `\s` : Whitespace
- `\S` : Not whitespace
- `\w` : Word
- `\W` : Not word
- `\d` : Digit
- `\D` : Not digit
- `\x` : Hexadecimal digit
- `\0` : Octal digit

Special Characters

- `\n` : Newline
- `\r` : Carriage return
- `\t` : Tab
- `\v` : Vertical tab
- `\f` : Form feed
- `\xxx` : Octal character xxx
- `\xhh` : Hex character hh

Groups and Ranges

- `.` : Any character except newline (\n)
- `(a|b)` : a or b
- `(…)` : Group
- `(?:…)` : Passive (non-capturing) group
- `[abc]` : a, b or c
- `[^abc]` : Not a, b or c
- `[a-z]` : Letters from a to z
- `[A-Z]` : Uppercase letters from A to Z
- `[0-9]` : Digits from 0 to 9

> Note: Ranges are inclusive.

Quantifiers

- `*` : 0 or more
- `+` : 1 or more
- `?` : 0 or 1
- `{3` : Exactly 3
- `{3,}` : 3 or more
- `{3,5}` : 3, 4 or 5

> Note: Quantifiers are greedy - they match as many times as possible. Add a ? after the quantifier to make it ungreedy.

Escape Sequences

- `\` :Escape following character. Used to escape any of the following metacharacters: {}^$.|*+?.
- `\Q` : Begin literal sequence
- `\E` : End literal sequence

String Replacement

- `$1` : 1st group
- `$2` : 2nd group
- `$n` : nth group
- `` $` ``: Before matched string
- `$'` : After matched string
- `$+` : Last matched string
- `$&` : Entire matched string

> Note: Some regex implementations use \ instead of $.

Assertions

- `?=` : Lookahead assertion
- `?!` : Negative lookahead
- `?<=` : Lookbehind assertion
- `?!=, ?<!` : Negative lookbehind
- `?>` : Once-only subexpression
- `?()` : Condition if-then

- `?()|` : Condition if-then-else
- `?#` : Comment

POSIX

- `[:upper:]` : Uppercase letters
- `[:lower:]` : Lowercase letters
- `[:alpha:]` : All letters
- `[:alnum:]` : Digits and letters
- `[:digit:]` : Digits
- `[:xdigit:]` : Hexadecimal digits
- `[:punct:]` : Punctuation
- `[:blank:]` : Space and tab
- `[:space:]` : Blank characters
- `[:cntrl:]` : Control characters
- `[:graph:]` : Printed characters
- `[:print:]` : Printed characters and spaces
- `[:word:]` : Digits, letters and underscore

Pattern Modifiers

- `g` : Global match
- `i` : Case-insensitive
- `m` : Multi-line mode. Causes ^ and $ to also match the start/end of lines.
- `s` : Single-line mode. Causes . to match all, including line breaks.
- `x` : Allow comments and whitespace in pattern
- `e` : Evaluate replacement
- `U` : Ungreedy mode

clarketm commented on Oct 25, 2017 • edited

Solid reference **@vitorbritto**!

loveplay1983 commented on Apr 11

Nice dude,i really need this.