



Regex almanac

22 words

PDF Created by **Create Dictionary**

20 October, 2024

Regex between 2 strings

(?<=(one two)).*?(?=four); one two three four ->three

(?<=^).*?(?= \$); returns whole string between start to finish

(?<=one).*?(?=four);one two three four->two three

Complete string if

^.*(person|server).*\$

This person is cool->this person is cool

vanil stick->nothing

server one->server one

contains non of these

^((?!.*(server|choco).*).)*\$

vanil stick->vanil stick

server one->nothing

surname (2nd word)

\\s+[^\s]+

mister meeseeks->meeseeks

double

\\d+(\\.\\d+)?

20.05

URL

[A-Za-z0-9.-]+\\.[A-Za-z]{2,6}

IPv4

([0-9].){4}[0-9]*

10.10.10.126

Phone example1

0\\d-\\d{7} for string 02-1234567 string string

^0\\d-\\d{7}\$

02-1234567

Duplicate word

\\b(\\w+)\\b(?=.*\\b\\1\\b) ; hadoken hadoken shoryuken->hadoken

\\b([\\w\\s']+) \\1\\b ; hadoken hadoken shoryuken->hadoken hadoken

Phone example 2

`[0]\d{9}` phone starting with 0 followed by 9 digits

Integer

`[-+]?[0-9]{1,13}`

Name Surname

`[A-Z][a-z]*\s[A-Z][a-z]*`

Heihachi Mishima

Negative positive double

`[-+]?[0-9]*[.][0-9]*`

-30.77

Full date

`[0-9]{1,4}/[0-9]{1,2}/[0-9]{1,2} [0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}`

2023/09/29 14:30:45

time stamp

`[0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2} ->11:52:31`

`[0-9]{1,2}:[0-9]{1,2}->11:42`

Email

`[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}`

`^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]*element[a-zA-Z0-9.-]*\.[a-zA-Z]{2,}$` ; email with element in it, good for bans

`^*@gmail.[a-zA-Z]{2,}$` ; emails from provider like gmail, good regex for bans

Any string

`^.+`

Metachars

`+` ; contains 1 or more of char

`*` ; contains 0 or more of char

`?` ; contains 0 or 1 of char

`(a group)` -> a group (exact match); `(abc)+`atches abc or abcabcab

`[]` ; class `[a-z]` lower case word, `[ab]` matches a or b

`{5}` ; quantifier; `ab{2}c` -> abbc

`|` ; or apple|banana matches: i ate an apple; i ate a banana

`.` ; any char: `a.c`-> abc a1c acc

`^` string start

`$` string end

Fancy numbers

`(0x)?[0-9a-fA-F]+` ; hex

`\\b[01]+\\b` ; binary

`(?<=0.).*` ; 0 stripper

`M{0,3}(CM|CD|D?C{0,3})(XC|XL|L?X{0,3})(IX|IV|V?I{0,3})` ; roman numerals

`[0-9]{1,3}(,[0-9]{3})*\\.[0-9]+` ; numbers with thousand separators

First word in a string

`^\\w+`

one two three-> one

Last word in a string

`\\w+$`

one two three four -> four

Number stripper

`[^\\d]+` ; h3llo->h llo