



RegEx

Regular Expression

Peyman Najafi

Quera

2021 April



Intro

- search pattern
- find or find and replace operations on strings
- in the 1950s by Stephen Cole Kleene
- common use with Unix text-processing utilities
- used in search engines, text processing utilities (sed, AWK)



Demo



Character Classes

[] : include character set

special characters `[abcDEF789 \t\\.\\(\\)\\[\\]\\{\\}]`

range of characters `[a-dG-P5-8]`

[^] : exclude character set

`[^abc]`

`[^a-h]`



Character Classes

\w : any alphanumeric : [a-zA-Z0-9_]

\W : any non-alphanumeric

\d : any digit : [0-9]

\D : any non-digit

\s : any whitespace : [\t\n\r\v\f]

\S : any non-whitespace

. : any character except line break

Anchor

`^...` : pattern start

`...$` : pattern end

```
/a[1-9]a/g  
a1a..a2a..a3a
```

```
/^a[1-9]a/g  
a1a..a2a..a3a
```

```
/a[1-9]a$/g  
a1a..a2a..a3a
```

Word Boundary

`\b` : word boundary

```
/\ba[1-9]a/g
```

```
zza1a..a2azz..zza3azz..a4a
```

```
/a[1-9]a/g
```

```
zza1a..a2azz..zza3azz..a4a
```

```
/a[1-9]a\b/g
```

```
zza1a..a2azz..zza3azz..a4a
```

```
/\ba[1-9]a\b/g
```

```
zza1a..a2azz..zza3azz..a4a
```

`\B` : not word boundary

Quantifiers

? : 0 or 1

* : 0 or more

+ : 1 or more

{m} : exactly m

{m,n} : between m and n

{m,} : m or more

```
/a1?a/g
```

```
aa • a1a • a11a
```

```
/a1*a/g
```

```
aa • a1a • a1111a
```

```
/a1+a/g
```

```
aa • a1a • a1111a
```

```
/a\d{2,4}a/g
```

```
a1a • a12a • a1234a • a12345a
```

match as few as possible (*? , +?)

```
/1[01]+1/g
```

```
1010111010100101011010101010100111
```

```
/1[01]+?1/g
```

```
1010111010100101011010101010100111
```




Groups

(ABC) : capturing group

(?<name>ABC) : named capturing group

python : (P?<name>ABC)

\1 \2 \3 : reference to captured group

Nested Groups

((ABC)(DEF))

```
/(?<path>(?!<directory>\\[\\w\\/\\.]*\\/)?(?!<name>[\\w\\.]*)\\.?!<extension>\\w+)/g
```

Text

Tests

NEW

File 1: /home/peyman/Downloads/video.mp4

File 2: /tmp/picture.1.png

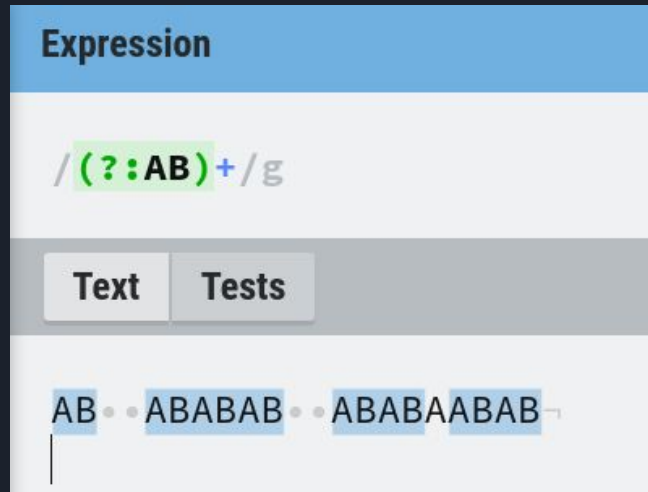
File 3: /tmp/folder.1/music.mp3

File 4: image.png

Demo

Groups

(?:ABC) : non-capturing group



Expression

```
/(?:AB)+/g
```

Text Tests

AB ABABAB ABABAABAB

The screenshot shows a regex testing interface. At the top, the expression `/(?:AB)+/g` is displayed, with the non-capturing group `(?:AB)` highlighted in green. Below the expression are two tabs: "Text" (selected) and "Tests". The "Text" tab shows the string `ABABABAABAB` with a vertical cursor at the beginning. Three matches are highlighted in light blue: `AB`, `ABABAB`, and `ABABAABAB`. The matches are separated by small grey dots, indicating the sequence of matches found by the regex engine.

Lookaround

(?=ABC) : positive lookahead

```
/\d(=?px)/g
```

```
1pt • 2px • 3em • 4px|
```

(?!ABC) : negative lookahead

```
/\d(?!px)/g
```

```
1pt • 2px • 3em • 4px|
```

(?<=ABC) : positive lookbehind

```
/(?<=age=)\d*/g
```

```
User(id=12, age=20)|
```

(?<!ABC) : negative lookbehind

```
/(?<!age=)\d{2}/g
```

```
User(id=12, age=20)|
```



Substitution

`$&` : all matched text

`$<id>` `$1` `$2` : captured group

`$`` : before matched text

`$'` : after matched text

Substitution Example

```
/(?<path>(?(<directory>\[/[\w\/\.\.]*\/)?(?<name>[\w\.\.]*)\.(?<extension>\w+))/g
```



Text

Tests NEW

4 matches (0.3ms)

```
File 1: /home/peyman/Downloads/video.mp4  
File 2: /tmp/picture.1.png  
File 3: /tmp/folder.1/music.mp3  
File 4: image.png
```

Tools

Replace

List

Details

Explain



```
<<.$3.$4>> is at <<.$2>>
```



```
File 1: << video.mp4 >> is at << /home/peyman/Downloads/ >>  
File 2: << picture.1.png >> is at << /tmp/ >>  
File 3: << music.mp3 >> is at << /tmp/folder.1/ >>  
File 4: << image.png >> is at << >>
```

Or : “|”

gr[e|a]y == gr[ae]y

```
/(\d{2})-(\d{3})-(\1|\2)/g
```

11-222-11 ↵

11-333-44 ↵

12-212-212 ↵

```
/(0|98|\+98)?(9[\d]{9}\b)/g
```

9123456789 ↵

09123456789 ↵

989123456789 ↵

+989123456789 ↵



Flags

g : global

i : ignore case sensitive

m : multi line

s : single line (dotall: "." include line break)



Persian characters

Specify with characters hex unicode : `[\u0000-\uffff]`

Persian characters hex unicode :

```
[\u0621-\u06cc\u00ab\u00bb\u060c\u061b]
```

```
/[\u0621-\u06cc\u00ab\u00bb\u060c\u061b]+/g
```

Persian RegEx Demo

Unicodelookup (find characters unicode)

- Chrome Regex Search extension Ctrl+Shift+F

- Google analytics

Goal description Edit
Name: *Conversion*
Goal type: *Destination*

2 Goal details

Destination

Regular expression Case sensitive

For example, use *My Screen* for an app and */thankyou.html* instead of *www.example.com/thankyou.html* for a web page.

`\\/college\\land\\college\\/(?<codeup>6090|7431).* /gm`

Text Tests NEW

`https://quera.ir/college/land/college/6090`

`https://quera.ir/college/land/college/2572`

`https://quera.ir/college/land/college/7431`

Python: re.findall



```
import re

pattern = r'\b\w\d+?\w\b'
text = 'a12b abcd ab x123y'
flags = re.M + re.I

re.findall(pattern, text, flags)

# ['a12b', 'x123y']
```

Python: re.search



```
import re

pattern = r'\b\w\d+?\w\b'
text = 'a12b abcd ab x123y'
flags = re.M + re.I

first_matched = re.search(pattern, text, flags)

first_matched.span() # (0, 4)

first_matched.group() # 'a12b'
```

Python: `re.match` (search from start)



```
import re

print(re.match(r'\w\d+\w', 'a22aaaa'))
# <re.Match object; span=(0, 4), match='a22a'>

print(re.match(r'\w\d+\w', 'aaa22a'))
# None
```

Python: re.sub

```
import re

pattern = r'(\d{4})-(\d{2})-(\d{2})'
replace_pattern = r'\1/\2/\3'
text = '''
2021-04-13
2021/05/15
2021-06-16
'''

re.sub(pattern, replace_pattern, text)

# 2021/04/13
# 2021/05/15
# 2021/06/16
```

Python: re.sub with replace function

```
import re

pattern = r'(\d{2,4})[-\/](\d{2})[-\/](\d{2})'

def fix_date(match: re.Match):
    # full_matched = match.group(0)
    year = match.group(1)
    if len(year) == 2:
        year = f'20{year}'
    month = match.group(2)
    day = match.group(3)
    return f'{year}/{month}/{day}'

text = '''
21-04-13
21/05/15
2021-06-16
'''

print(re.sub(pattern, fix_date, text))

# 2021/04/13
# 2021/05/15
# 2021/06/16
```

Javascript

/pattern/flags

text.match(pattern)



```
const pattern = /\w\d+\w/g;  
const text = "aa b1b c12c";
```

```
let result = text.match(pattern)  
console.log(result)
```

```
// ["b1b", "c12c"]
```


Javascript

text.matchAll(pattern) for more details



```
const pattern = /(\w)(\d+)(\1)/g;  
const text = "aa b1b c12c";  
  
let matchAll = text.matchAll(pattern)  
matchAll = Array.from(matchAll)  
console.log(matchAll)
```

```
▼ (2) [Array(4), Array(4)] ⓘ  
  ▼ 0: Array(4)  
    0: "b1b"  
    1: "b"  
    2: "1"  
    3: "b"  
    groups: undefined  
    index: 3  
    input: "aa b1b c12c"  
    length: 4  
    ▶ __proto__: Array(0)  
  ▼ 1: Array(4)  
    0: "c12c"  
    1: "c"  
    2: "12"  
    3: "c"  
    groups: undefined  
    index: 7  
    input: "aa b1b c12c"  
    length: 4  
    ▶ __proto__: Array(0)  
length: 2  
▶ __proto__: Array(0)
```

Javascript

text.split(pattern)



```
let text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit; sed do eiusmod tempor incididunt  
(ut labore et dolore magna aliqua) Ut enim ad minim veniam.";
```

```
text.split(/ ?[\.,;\(\)]+ ?/);
```

```
/*  
0: "Lorem ipsum dolor sit amet"  
1: "consectetur adipiscing elit"  
2: "sed do eiusmod tempor incididunt"  
3: "ut labore et dolore magna aliqua"  
4: "Ut enim ad minim veniam"  
5: ""  
*/
```

Javascript

text.replace(str|regexp, str|func)



```
"abc-def-gh".replace("-", " ", " ") // "abc, def-gh"
```

```
"abc-def-gh".replace(/-/g, " ", " ") // "abc, def, gh"
```

```
"2021/19/04, 2021/29/04".replace(/(\d{4})\/(\d{2})\/(\d{2})/g, "$1/$3/$2")  
// "2021/04/19, 2021/04/29"
```

```
function fixDate(match, year, day, month) {  
  year = year.length === 2? `20${year}`: year;  
  return `${year}/${month}/${day}`  
}
```

```
"21/19/04, 2021/29/04".replace(/(\d{2,4})\/(\d{2})\/(\d{2})/, fixDate)  
// "2021/04/19, 2021/29/04"
```

```
let text = `
Quera started in the summer of 2015 with 3 member.
College of quera started in the spring of 2019 with 5 members.
Codeup started in the summer of 2020 with 8member.
`;
let regexp = /([\w ]+)(?: started in the )(.*)?(?: of )(\d+)(?: with )(\d+)(?: *members?)/igm;

let result;

while (result = regexp.exec(text)) {
  console.log(`${result[3]} ${result[2]}, ${result[1]}, ${result[4]} member | position
${result.index}`);
}

/*
2015 summer, Quera, 3 member | position 1
2019 spring, College of quera, 5 member | position 52
2020 summer, Codeup, 8 member | position 116
*/
```



grep

print lines matching a pattern

```
grep [OPTIONS] PATTERN [FILE...]
```

```
grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]
```



grep

-P “pattern”

-r : recursively read all files under each directory

-i : Ignore case sensitive

-c : count of matching lines for each file

-o : only matched section

-n : line number

-w : search in words

grep

```
regex/grep_files [ grep -rn -P "\b\w+era" ./*  
./dir1/file3.txt:2:Here is quera  
./dir1/file3.txt:5:Quera is developers community  
./file1.txt:1:quera  
./file1.txt:3:Quera  
./file2.txt:3:Coursera  
regex/grep_files [ grep -rno -P "\b\w+era" ./*  
./dir1/file3.txt:2:quera  
./dir1/file3.txt:5:Quera  
./file1.txt:1:quera  
./file1.txt:3:Quera  
./file2.txt:3:Coursera  
regex/grep_files [ grep -rc -P "\b\w+era" ./*  
./dir1/file3.txt:2  
./file1.txt:2  
./file2.txt:1
```

grep

```
grep_files: zsh -- Konsole
File Edit View Bookmarks Settings Help
79.136.114.202 - - [04/Jun/2015:07:06:29 +0000] "GET /downloads/product_1 HTTP/1.1" 404 319 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.22)"
80.91.33.133 - - [04/Jun/2015:07:06:14 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.22)"
79.136.114.202 - - [04/Jun/2015:07:06:52 +0000] "GET /downloads/product_1 HTTP/1.1" 200 490 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.22)"
79.136.114.202 - - [04/Jun/2015:07:06:50 +0000] "GET /downloads/product_1 HTTP/1.1" 404 334 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.22)"
80.91.33.133 - - [04/Jun/2015:07:06:16 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.16)"
80.91.33.133 - - [04/Jun/2015:07:06:30 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.16)"
192.235.75.62 - - [04/Jun/2015:07:06:45 +0000] "GET /downloads/product_1 HTTP/1.1" 404 331 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.12)"
141.138.90.60 - - [04/Jun/2015:07:06:46 +0000] "GET /downloads/product_2 HTTP/1.1" 200 3316 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.15)"
141.138.90.60 - - [04/Jun/2015:07:06:31 +0000] "GET /downloads/product_2 HTTP/1.1" 200 490 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.15)"
173.255.199.22 - - [04/Jun/2015:07:06:04 +0000] "GET /downloads/product_2 HTTP/1.1" 404 339 "-" "Debian APT-HTTP/1.3 (0.8.10.3)"
54.186.10.255 - - [04/Jun/2015:07:06:05 +0000] "GET /downloads/product_2 HTTP/1.1" 200 2582 "-" "urlgrabber/3.9.1 yum/3.4.3"
80.91.33.133 - - [04/Jun/2015:07:06:16 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.16)"
144.76.151.58 - - [04/Jun/2015:07:06:05 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
79.136.114.202 - - [04/Jun/2015:07:06:35 +0000] "GET /downloads/product_1 HTTP/1.1" 404 334 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.22)"
regex/grep_files [ grep -P "^\\d+\\.\\d+\\.\\d+\\.\\d+" nginx_logs ] 1:57 PM
```


grep

```
grep_files : zsh — Konsole
File Edit View Bookmarks Settings Help
20 199.68.238.7
20 208.113.156.25
20 209.216.233.52
20 212.83.187.74
20 216.46.173.126
20 216.46.173.126
20 217.64.170.250
20 217.64.170.250
20 84.208.15.12
20 94.75.199.161
21 136.243.233.130
21 144.76.156.176
22 87.242.74.154
23 119.252.76.162
23 184.106.13.229
23 78.100.89.194
24 162.221.226.158
24 193.30.60.25
25 74.125.60.158
26 209.216.233.52
27 195.145.19.12
30 119.252.76.162
30 147.203.99.20
30 173.11.48.149
30 216.46.173.126
30 54.165.228.106
30 74.125.60.158
36 74.125.60.158
regex/grep_files [ grep -o -P "^\d+\.\d+\.\d+\.\d+" nginx_logs | uniq -c | sort -n █ ] 2:01 PM
```

grep

```
896 04/ Jun/2015
1966 17/May/2015
2825 03/ Jun/2015
2831 19/May/2015
2836 29/May/2015
2837 01/ Jun/2015
2839 25/May/2015
2851 20/May/2015
2852 28/May/2015
2853 24/May/2015
2855 18/May/2015
2863 31/May/2015
2864 02/ Jun/2015
2876 30/May/2015
2879 22/May/2015
2879 26/May/2015
2881 21/May/2015
2887 27/May/2015
2892 23/May/2015
```

```
regex/grep_files [ grep -o -P "\d{2}\/\w{3}\/\d{4}" nginx_logs | uniq -c | sort -n █
```

```
] 2:07 PM
```



Resources

- [regexr.com](#)
- [regexone.com](#)
- [Python re](#)
- [Javascript regex](#)
- [grep](#)
- [regex-google-analytics](#)