Regular Expressions

Prabhjeet Singh

Regular Expressions

Searching with Regular Expressions

There are 4 primary components of regular expressions.

1. Character classes → in [] brackets e.g. – [A-z], [0-9]
2. Quantifiers and alternation → +, *, ? -> quantifiers → {1,3} min or max value matched
3. Groups → ()
4. Anchors → where match starts and ends.

Example – valid date expression -> month, date, year

^[A-Z][a-z]{2,}\s+[0-3]?[1-9],\s+[12]?[0-9]{0,3}$

Example – to find number is 42 or not

^4[2-9]|[5-9]\d|[1-9]\d{2,}$

Character classes	
.	any character except newline
\w\d\s	word, digit, whitespace
\W\D\S	not word, digit, whitespace
[abc]	any of a, b, or c
[^abc]	not a, b, or c
[a-g]	character between a & g
Anchors	
^abc$	start / end of the string
\b\B	word, not-word boundary
Escaped characters	
\.*\\	escaped special characters
\t\n\r	tab, linefeed, carriage return
Groups & Lookaround	
(abc)	capture group
\1	backreference to group #1
(?:abc)	non-capturing group
(?=abc)	positive lookahead
(?!abc)	negative lookahead
Quantifiers & Alternation	
a*a+a?	0 or more, 1 or more, 0 or 1

a{5}a{2,}	exactly five, two or more
a{1,3}	between one & three
a+?a{2,}?	match as few as possible
ab|cd	match ab or cd

In Terminal

└─$ cat number.txt

7

33

41

42

55

100

1000

└─$ grep -E '^4[2-9]|[5-9]\d|[1-9]\d{2,}$' number.txt

42

Expression is single quotes

Use -E for the expression

Grep is to search

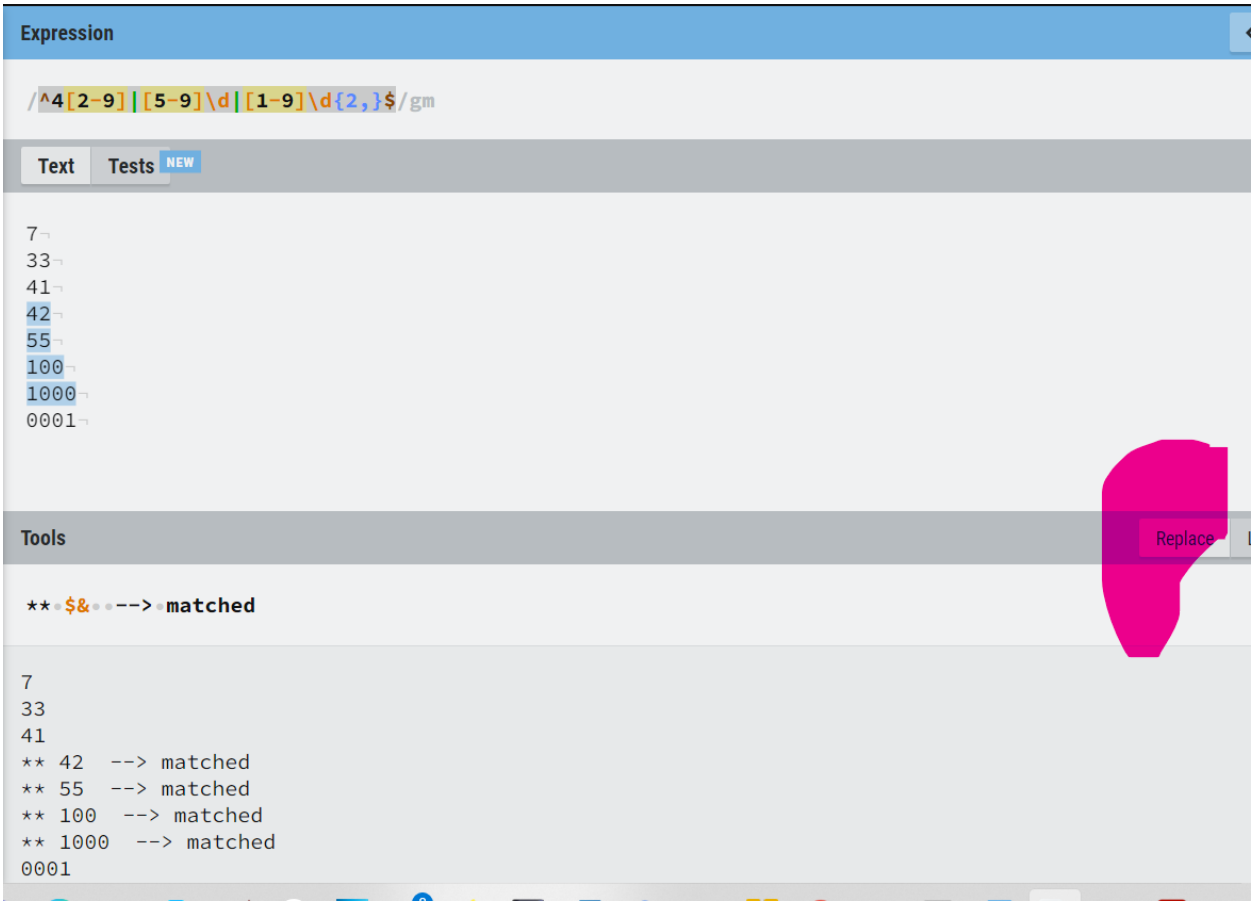
And number.txt for input numbers.

Other examples –

```
bob@linux101:~$ cat numbers.txt
7
33
41
42
55
100
1000
bob@linux101:~$ grep -E '^4[2-9]|[5-9]\d|\d{3,}$' numbers.txt
42
bob@linux101:~$ grep -E '^4[2-9]|[5-9][0-9]|[0-9]{3,}$' numbers.txt
42
55
100
1000
bob@linux101:~$ grep -E '^4[2-9]|[5-9][[:digit:]]|[[:digit:]]{3,}$' numbers.txt
42
55
100
1000
bob@linux101:~$
```

Replacing texts with Regular Expressions.

Use $& or $1 , $2 etc to replace the matching value with any value.



Tips on Building Own Regular Expressions

1. Regular expressions are greedy
    a. Add an ? after * or + to make it lazy
2. Don't build an expression all at once
    a. Build a piece of it and test it and then repeat it.
    b. Use multiple, simpler expressions
3. Test valid and invalid data
4. Add comments using x modifier.