RegEx cheatsheet (GNU BRE/ERE)[1]

Tom Hinne

March 12, 2025

Pattern Matching Tools

grep → print lines that match patterns
  -E → enable extended regex (ERE)
  -F → interpret as fixed strings, not regex
  -i → enable case-insensitive search
  -c → print count of matching lines
  -q → suppress output display, return exit
    code 0 if match found
  -v → invert match
egrep → identical to grep -E
sed → stream editor for filtering and transforming text
  -r → enable extended regex (ERE)
awk → pattern searching and processing language

Basic Regular Expression (BRE)

`.` → any characters

`*` → zero or more occurrences of the preceding character or group

`^` → start of a string or line

`$` → end of a string or line

`[]` → any character within the specified set or range

`[^]` → any character **except** those listed inside the brackets

`\` → nullifies the special meaning of the next character

POSIX Character Classes

`[:alnum:]` → digits, upper- and lowercase letters

`[:word:]` or `\w` → [:alnum:] and underscore

`[:alpha:]` → upper- and lowercase letters

`[:blank:]` → space and TAB

`[:cntrl:]` → control characters

`[:graph:]` → graphical characters (invert of [:cntrl:])

`[:print:]` → [:graph:] and space

`[:lower:]` → lowercase letters

`[:punct:]` → punctuation

`[:space:]` or `\s` → blank characters

`[:upper:]` → uppercase letters

`[:xdigit:]` → hexadecimal digits

`[:digit:]` → digits

`[abc]` → characters a or b or c

`[a-f]` → characters a,b,c,d,e,f

`[^abc1]` → any characters except a or b or c or 1

`[abc[:digit:]]` → digits and a or b or c

Extended Regular Expression (ERE)

`()` → group expressions or capture matching substrings

`|` → choice between expressions

`?` → 0 or 1 occurrence of the preceding character or group

`+` → 1 or more occurrences of the preceding character or group

`{}` → specify a precise number of occurrences of the preceding character or group
  `a{2,4}` → aa or aaa or aaaa
  `ab{3}` → abbb
  `ab{3,}` → abbb or abbbb, or abbbbb, or ...
  `ab{,3}` → a or ab or abb or abbb

Word boundaries

`\b` or `\<...>\` → specifies word beginning or end

`\B` → does NOT specifies word beginning or end

Backreferences

`\1` through `\9` → refer to a previously captured group

Examples

HTTP or HTTPS URL :
`^https?://[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`

match word "progra":
`\bprogra\b`

US phone number: (XXX-XXX-XXXX)
`^[0-9]{3}-[0-9]{3}-[0-9]{4}$`

Belgian zip code: (from 1000 to 9992)
`^[1-9][0-9]{3}$`

Date: (DD-MM-YYYY)
`^(0[1-9]|[12][0-9]|3[01])-(0[1-9]|1[0-2])-[0-9]{4}$`

IPv4 address:
`^(\b25[0-5]|\b2[0-4][0-9]|\b[01]?[0-9][0-9]?)(\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)){3}$`

References

[1] "GNU Grep 3.7," www.gnu.org.
https://www.gnu.org/software/grep/manual/grep.html

[2] "sed, a stream editor," www.gnu.org.
https://www.gnu.org/software/sed/manual/sed.html