

Using Regular Expressions

REGEX

Pattern matching for a certain amount of text

Single character: O

`Odybot isn't human`

Character sets: [a-z]

`Odybot isn't human`

Character sets: [aei]

`Odybot isn't human`

Character sets: [0-9]

`Odybot isn't human`

Non printable characters:

`\t` : tab

`\r` : carriage return

`\n` : new line (Unix)

`\r\n` : new line (Windows)

`\s` : space

Special Characters:

`.` period or dot: match any character

`\` backslash: make next character literal

`^` caret: matches at the start of the line

`$` dollar sign: matches at the end of line

`*` asterisk or star: repeat match

`?` question mark:

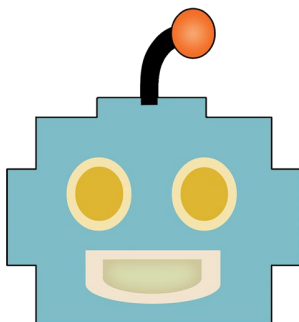
`+` plus sign:

`()` parentheses: create a capturing group

`[]` square bracket: sequence of characters

also seen like `[[:name:]]` or `[[.az.]]`

`{ }` curly brace: place bounds



38 Oxford St, Rm 206
rc.fas.harvard.edu
rchelp@fas.harvard.edu
odybot.rc.fas.harvard.edu
Office hours: M-F, 12-3p

GNU regular expression parser: **grep**

a line by line parser of stdin and by default displays matching lines to the regex pattern

Syntax:

using stdin: `cat file | grep pattern`

using files: `grep pattern file`

Common Options:

`-c` : count the number of occurrences

`-m #` : repeat match # times

`-R` : recursively through directories

`-o` : only print matching part of line

`-n` : print the line number

`-v` : invert match, print non-matching lines

Stream editor: **sed**

takes a stream of stdin and pattern matches and returns to stdout the replaced text.

Syntax:

using stdin: `cat file | sed 'command'`

using files: `sed 'command' file`

Command Common Uses:

`4d` : delete line 4

`2,4d` : delete lines 2-4

`2w foo` : write line 2 to file foo

`/here/d` : delete line matching here

`/here/,there/d` : delete lines here to there

`s/pattern/text/` : switch text matching first occurrence of pattern

`s/pattern/text/g` : switch text matching pattern globally

`/pattern/a/text` : append line with text after matching pattern

`/pattern/c/text` : change line with text for matching pattern

Common Option:

`-i` : edit file in place

Example:

`sed 's/abc/123/' abc.txt`