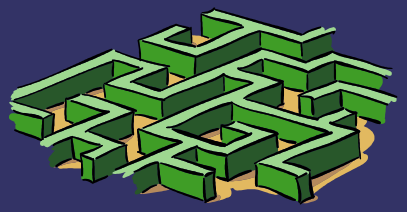
GREP – your new best friend (sort of)



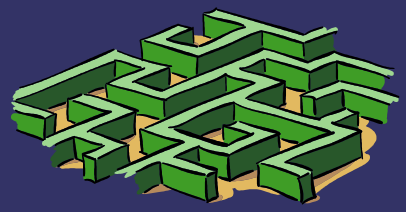
The Abridged History of grep

Your new best friend's step-dad: *ed*

grandpa: Ken Thompson in 1971

Why is *ed* cooler than the other dads?

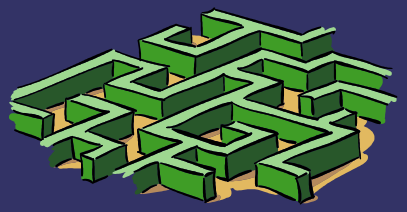
Implementation of RegEx



RegEx, you say?

/Everybody stand back/

I know regular expressions

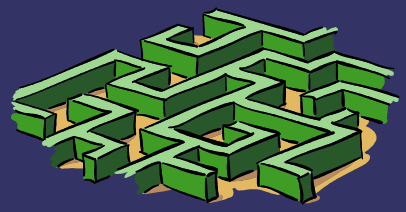


Wikipedia Says:

Regular Expression:

In computing, a regular expression provides a concise and flexible means to "match" (specify and recognize) strings of text, such as particular characters, words, or patterns of characters.

- Simple Example: [hc]at matches "hat" and "cat".
- CRAZY Example: a valid email address RegEx



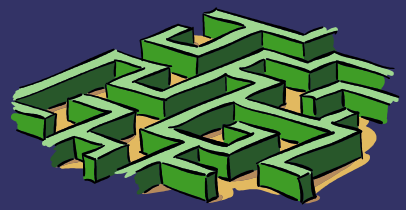
- 6,343 characters.

Holy Shit Balls

Why do I care?

- RegEx are efficient
- RegEx are ubiquitous

- grep = g/re/p = (global / regular expression / print)

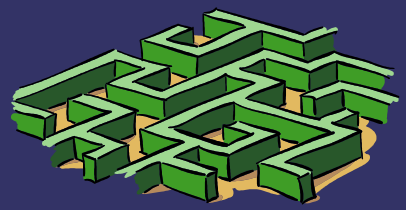


grep = g/re/p = (global / regular expression / print)

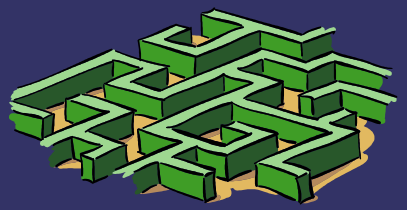
Wikipedia:

“grep is a command-line utility for searching plain-text data sets for lines matching a regular expression.”

- g/re/p (global / regular expression / print) is an *ed* command.



So... what does that mean?

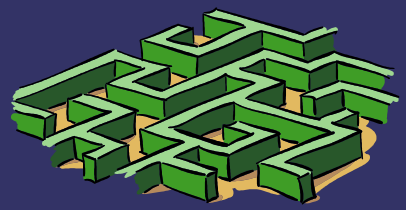


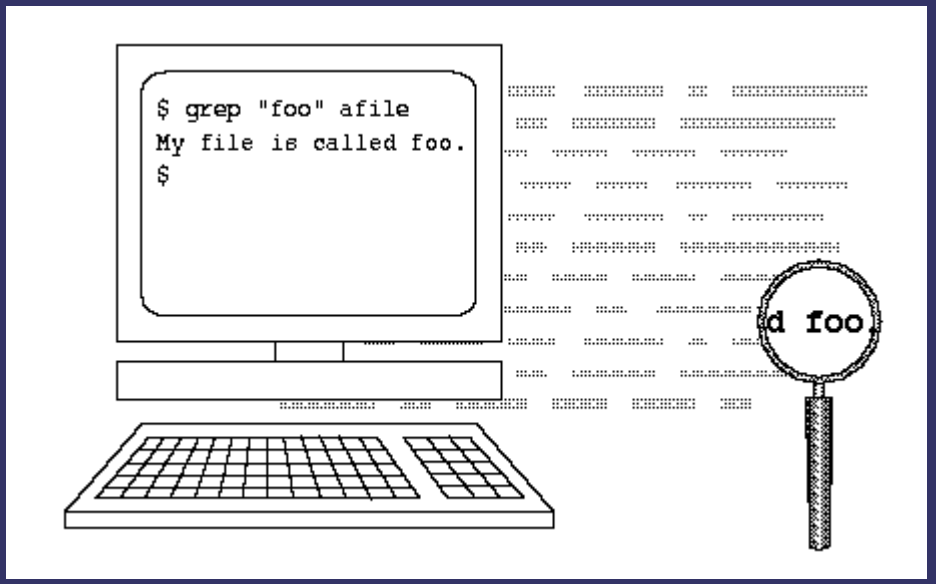
Put Simply

- Grep searches files.

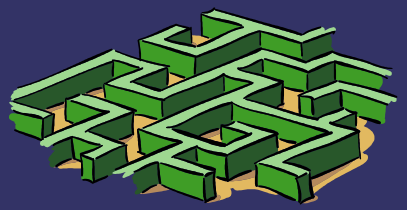
- You tell grep:
  - What you are looking for
  - Where you are looking for it
  - How print that Motherf****er out.

It's like the “search” function of your browser/editor




$ grep "foo" afile
My file is called foo.
$
d foo


FINDING A
IN A
= SUCCESS!!
shenkitup.com



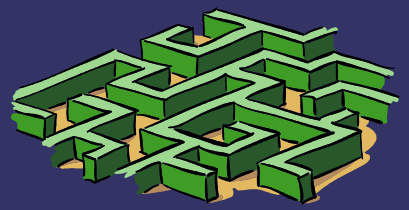
Examples – Basic
The not-so-forbidden fruit

fruitlist.txt

- apple
- apples
- pineapple
- apple-
- apple-fruit
- fruit-apple

- grep apple fruitlist.txt

- apple
- apples
- pineapple
- apple-
- apple-fruit
- fruit-apple



Examples – Basic pt. 2
this and only this

fruitlist.txt

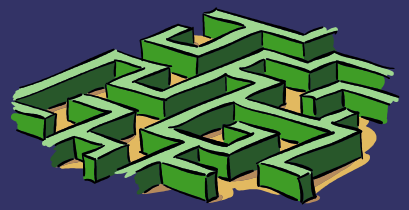
- apple
- apples
- pineapple
- apple-
- apple-fruit
- fruit-apple

- grep -x apple fruitlist.txt

- apple

Exact line match is performed with the option flag x.

Lines only containing exactly and solely apple are selected with a line-regexp instead of word-regexp:



Examples – Funky Stuff
anti-apples

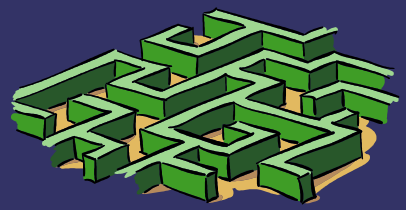
fruitlist.txt

- apple
- apples
- pineapple
- apple-
- apple-fruit
- Fruit-apple
- banana
- pear
- peach
- orange

- grep -v apple fruitlist.txt

- banana
- pear
- peach
- Orange

The v (lower-case V) option reverses the sense of the match and prints all lines that do not contain apple, as in this example.

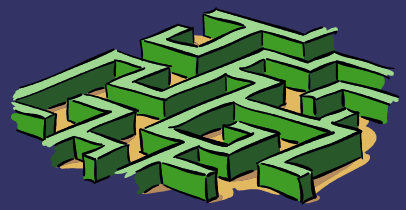


A Practical Example

- You are a having a problem with the Koans
- You remember the Koan had something to do with “eins”
- You can't remember which Koan had all that stupid “eins” nonsense

grep eins *.rb

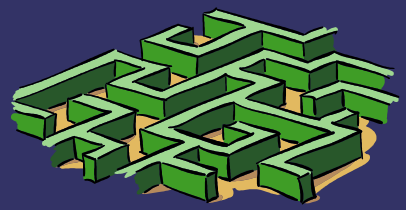
- “show me where *eins* appears in all ruby files in my pwd”
- Will shit the bed if you have no .rb files in your pwd



*What does “grep eins *.rb” output?*

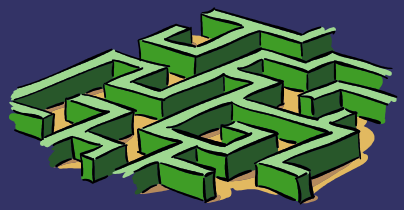
```
about_hashes.rb:    hash[:one] = "eins"
about_hashes.rb:    expected = { :one => "eins", :two => "dos" }
```

^^ where it is ^^what it is



Grep options

- The options in your greps are vast:
  - You should just look them up as you need them
  - The useful ones will become second nature
    EG:
  - ls -1 | grep ^a | wc -l
    - Count the files in the current directory that begin with the letter a.
      To prevent confusion, the above commands reads ls <dash><the #1> <pipe> grep <carrot>a <pipe> wc <dash><the letter l>. What this does is use the ls command to list files in a bare format, use grep to filter only the files beginning with a, and then pipe that output into the wc command to count how many files are listed. When done properly the terminal should return a single number and then the prompt.

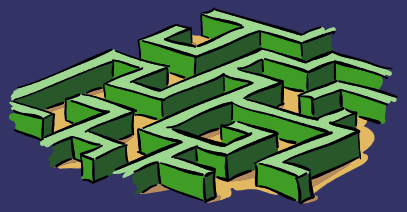


Fun Fact! (via Wikipedia)

In December 2003, the Oxford English Dictionary Online added draft entries for "grep" as both a noun and a verb.

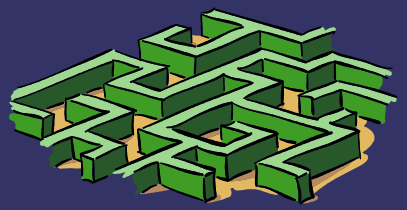
A common verb usage is the phrase "You can't grep dead trees"—meaning one can more easily search through digital media, using tools such as grep, than one could with a hard copy (i.e., one made from dead trees, paper).

Google is a verb, right?



Resources

- http://rubular.com/
- http://en.wikipedia.org/wiki/Grep
- http://www.computerhope.com/unix/ugrep.htm



감사합니다 Natick

Grazie Danke Ευχαριστίες Dalu Obrigado

Thank You Köszönöm

Tack

Спасибо Dank Gracias

谢谢 Merci Seé

ありがとう