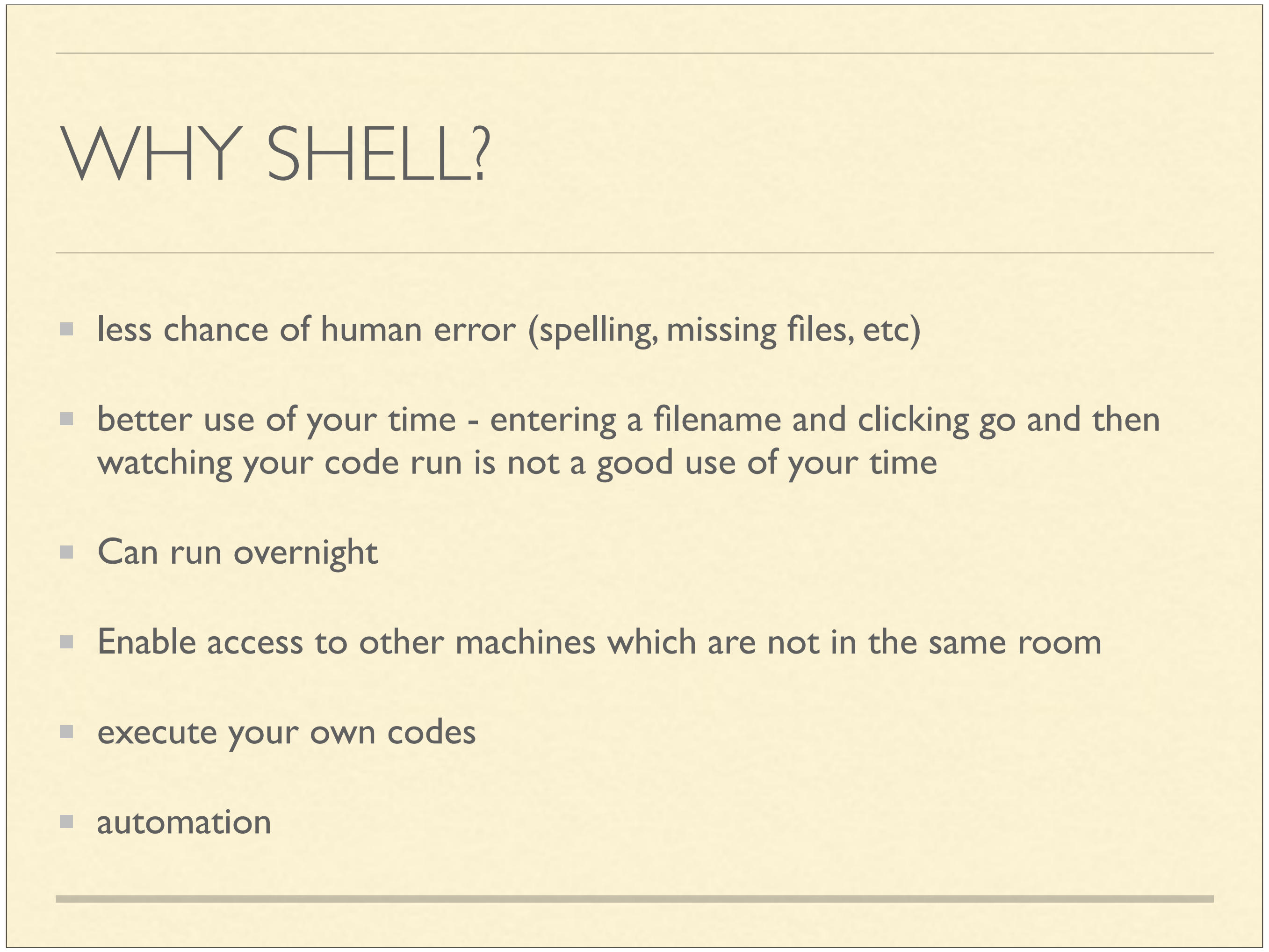
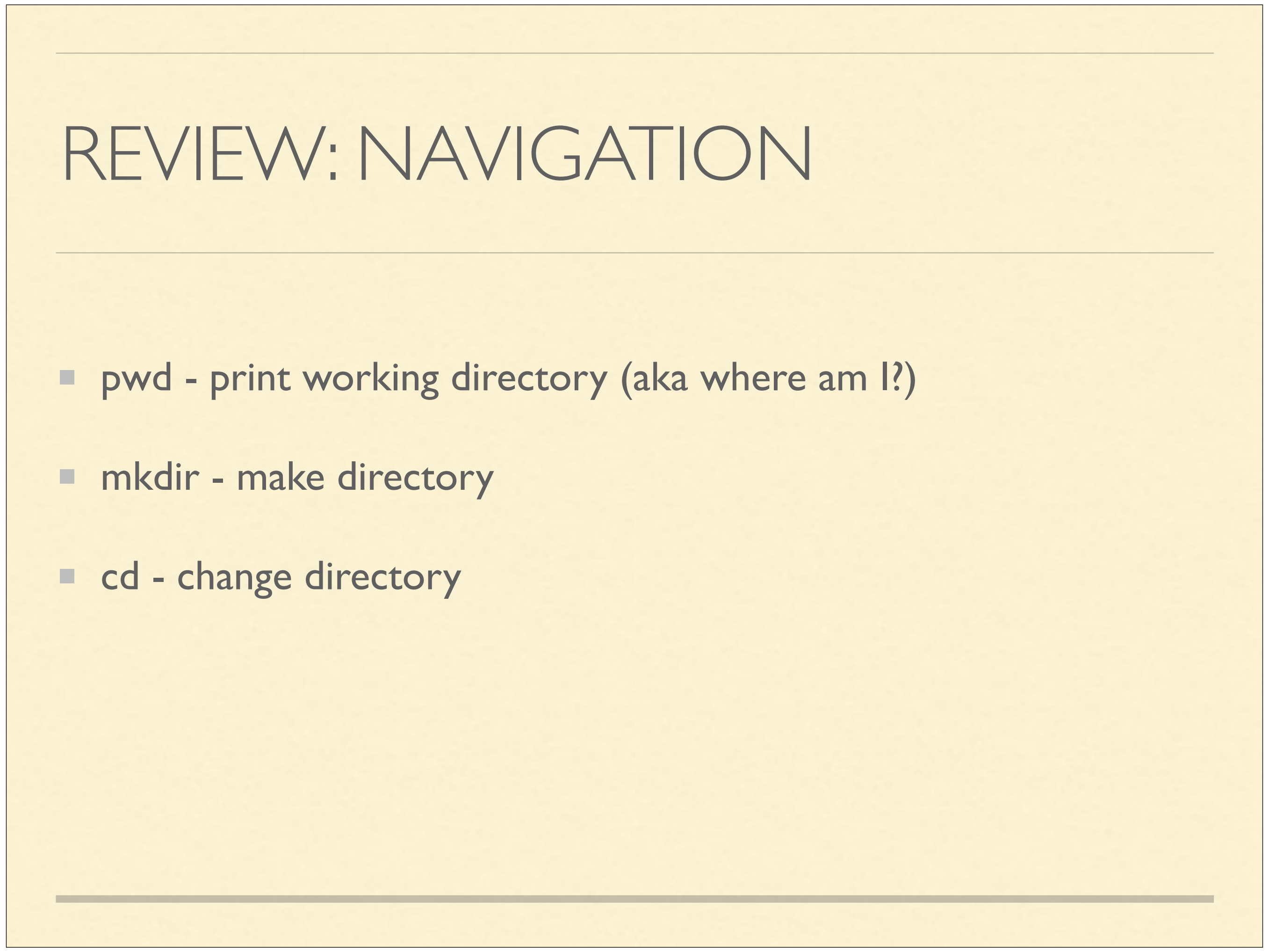
# SHELL



## WHY SHELL?

- less chance of human error (spelling, missing files, etc)
- better use of your time - entering a filename and clicking go and then watching your code run is not a good use of your time
- Can run overnight
- Enable access to other machines which are not in the same room
- execute your own codes
- automation

More than you know is possible



## REVIEW: NAVIGATION

- pwd - print working directory (aka where am I?)
- mkdir - make directory
- cd - change directory

Macs are not case sensitive

## GET FILES FOR COURSE

- Decide where you want your files to go
- cd there
- git clone …
- put up your green post-it when you are done

## REVIEW: FILE MANAGEMENT

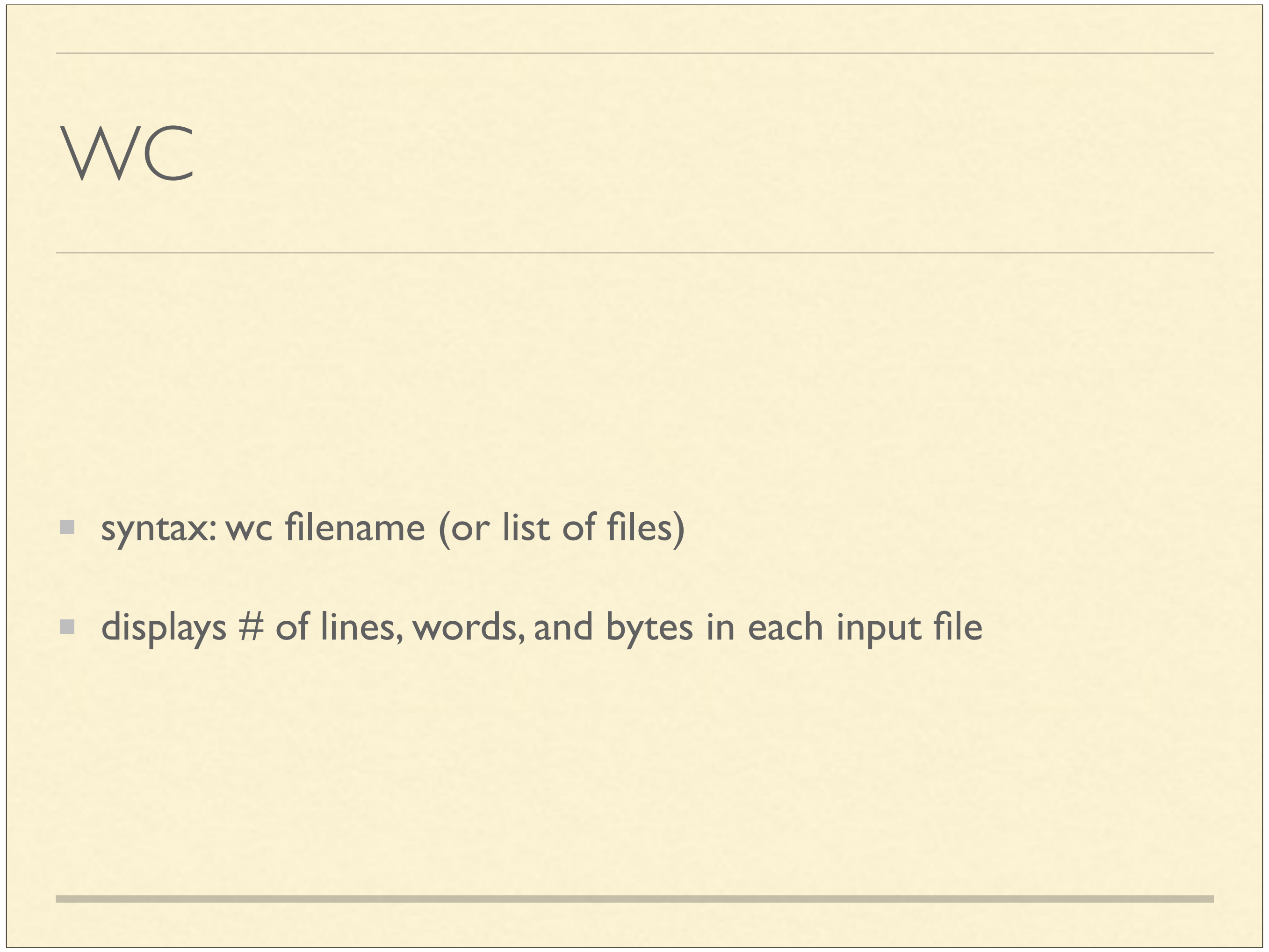
- ls - list what is in my directory
    - can pass parts of a path or full path to narrow results
- touch filename - update or create empty file
- cat filename - print file contents to screen
- cp from_path/filename to_path/filename- copy file
- mv from_path/filename to_path/filename- move file
- rm path/filename - remove file
- *, ? - wildcards

## EXERCISE:

- cd into 2014-04-14-wise/intermediate/shell
- make a directory called shell_sandbox
- inside shell_sandbox make a directory called molecules
- copy all .pdb files from 2014-04-14-wise/novice/shell/molecules into molecules
- put up your green post-it

## BEYOND BASIC NAVIGATION

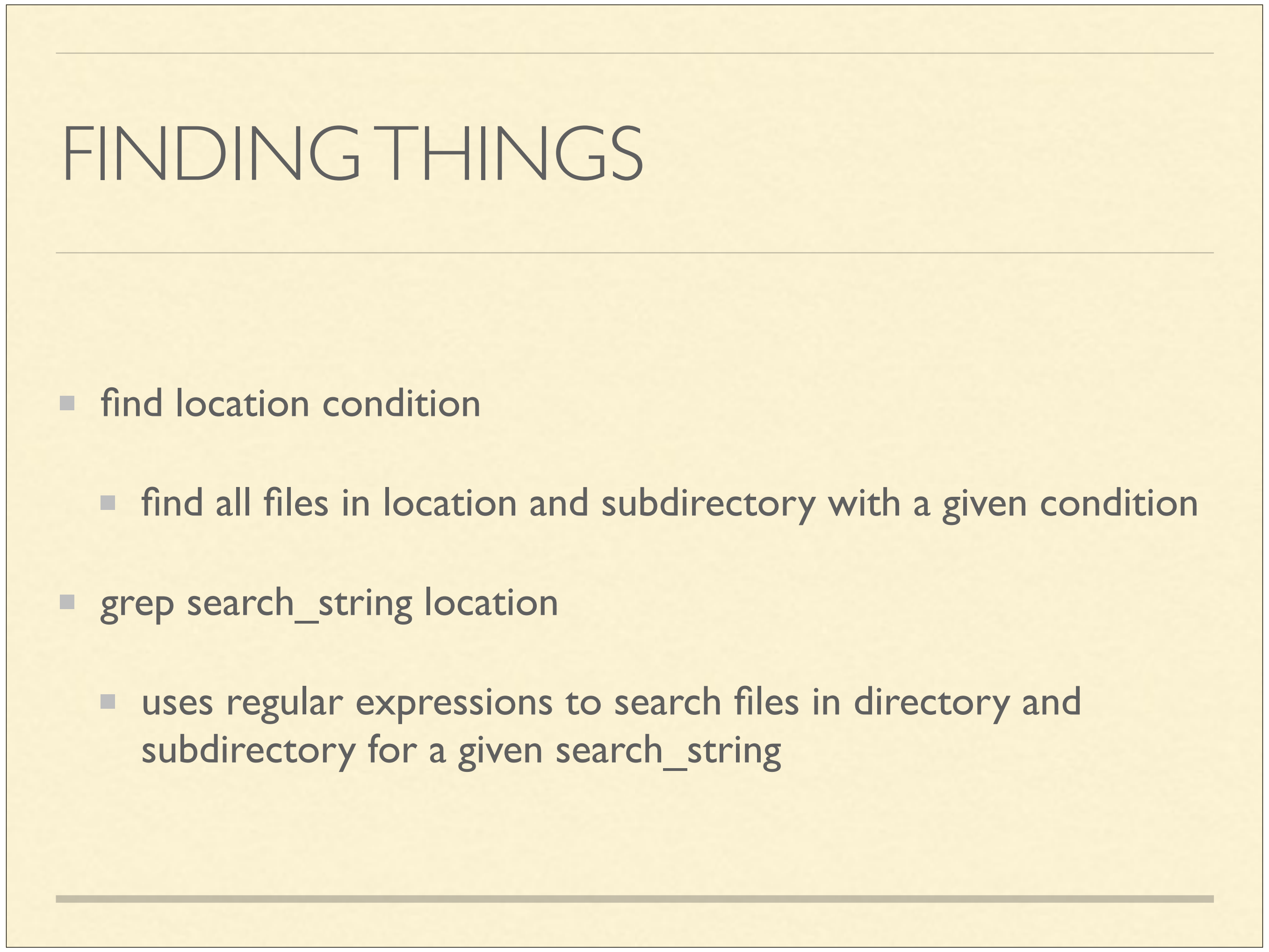
- cd 2014-04-14-wise folder
- man command - get help on command
- Exercise:
  - use man to figure out what wc does



## WC

- syntax: wc filename (or list of files)
- displays # of lines, words, and bytes in each input file

```
wc novice/shell/molecules/cubane.pdb
    20    156    1158 novice/shell/molecules/cubane.pdb
```



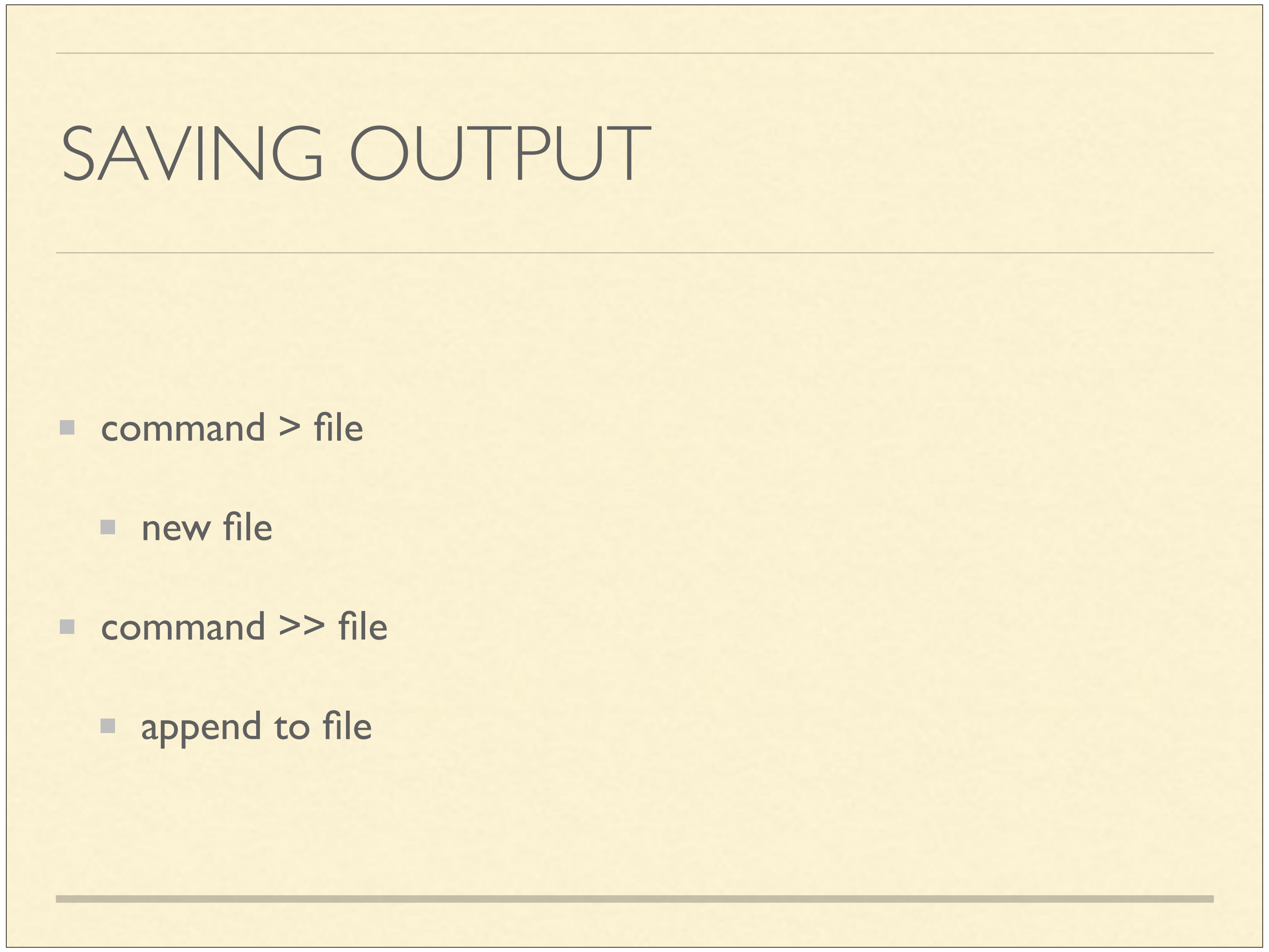
## FINDING THINGS

- find location condition
  - find all files in location and subdirectory with a given condition
- grep search_string location
  - uses regular expressions to search files in directory and subdirectory for a given search_string

examples: find -name ./  ../../*.pdb

Lots of options

example: grep shell *.md



## SAVING OUTPUT

- command > file
  - new file
- command >> file
  - append to file

Example: ls intermediate/shell/shell_sandbox/molecules/*.pdb > molecule_filelist.txt

Example ls intermediate/shell/shell_sandbox/molecules/*.pdf >> molecule_filelist.txt

## COMBING COMMANDS

- Shell is most powerful when you can use it to string commands together. How do you pass the output of one command as input to another?
- | (pipe)
- How many times does the word shell occur in .md files?

```
grep shell *.md | wc
```

## EXERCISE:

- Use find to get the number of files you copied into the intermediate/shell/shell_sandbox/molecules folder

Solution: find intermediate/shell/shell_sandbox/molecules -name *.pdb

## SCRIPTING

- What if you want to do something more than once
  - write a script and save it

## PERMISSIONS: WHAT DOES IT MEAN?

- cd intermediate/shell/sandbox
- ls -l
- drwxrwxrwx owner group size last_updated filename
  - owner (user), group, other
  - r = read
  - w = write
  - x = execute

## CHANGING PERMISSIONS

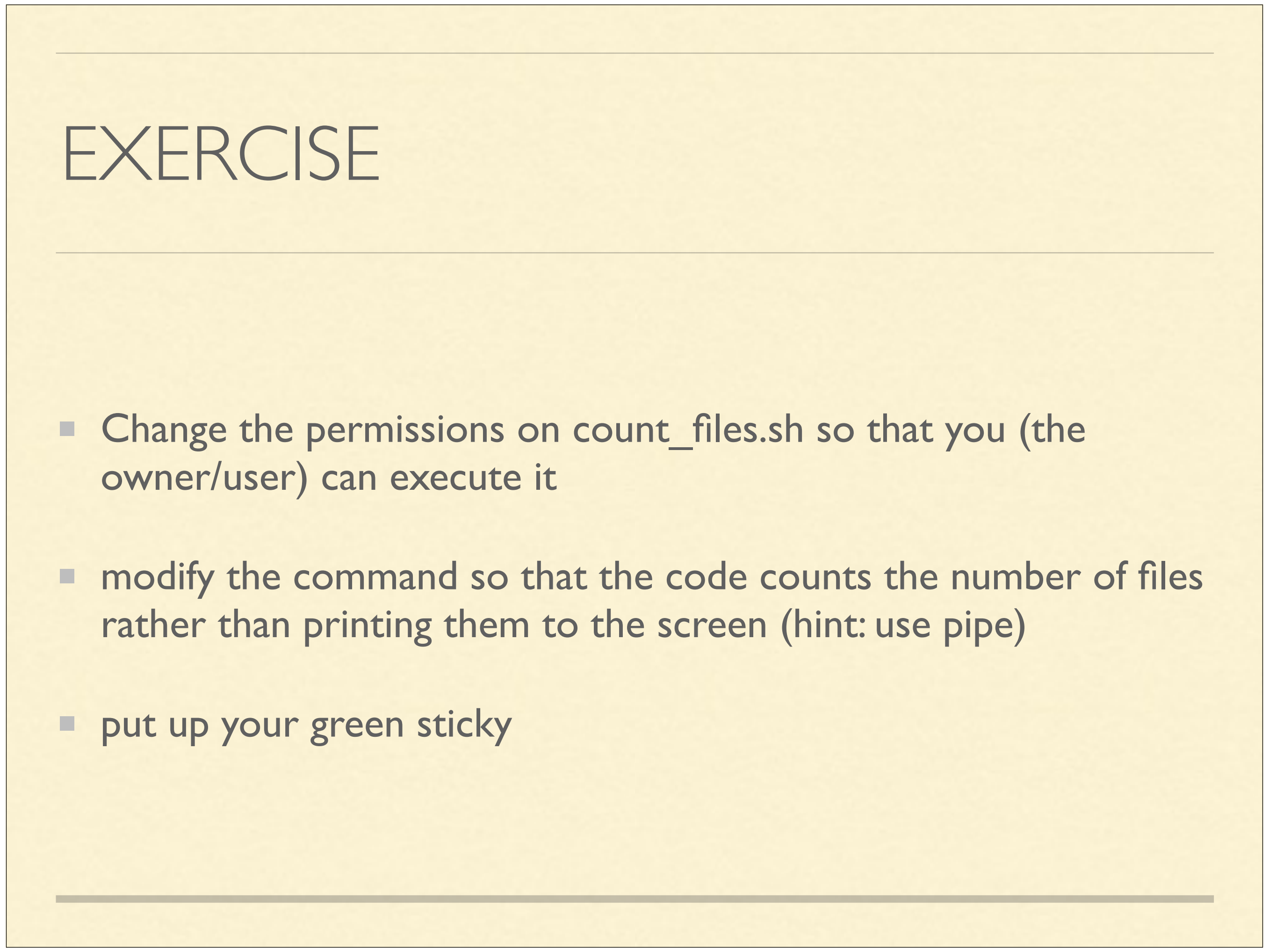
- chmod who+/-what
  - example:
    - cd into shell_sandbox
    - touch count_files.sh
    - ls -l count_files.sh
    - chmod g+w count_files.sh

## EDITOR: NANO

- In the command line
- barebones basic
- works everywhere
- most important commands at the bottom of the screen
- ^ = control
- start by typing nano

## WRITE A SCRIPT

- put in count_files.sh file
  - ls -l molecules/*.pdb
- Execute script:
  - always work:
    - sh count_files.sh
  - try:
    - ./count_files.sh
    - permission error

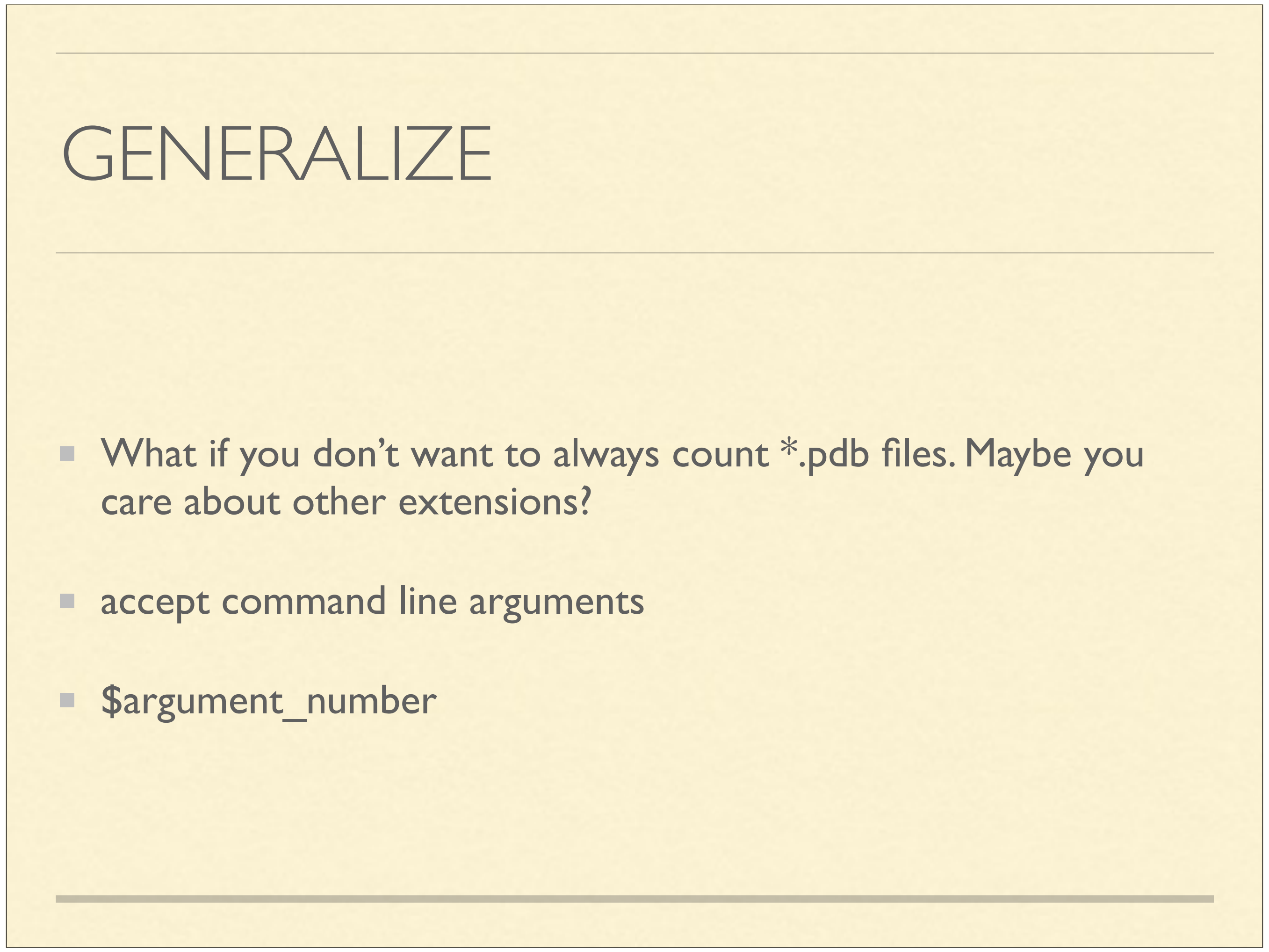


## EXERCISE

- Change the permissions on count_files.sh so that you (the owner/user) can execute it
- modify the command so that the code counts the number of files rather than printing them to the screen (hint: use pipe)
- put up your green sticky

```
!# /bin/bash


ls -l molecules/*.pdb | wc
```



## GENERALIZE

- What if you don’t want to always count *.pdb files. Maybe you care about other extensions?
- accept command line arguments
- $argument_number

Expanded before passed in, so $1 is the only the first item in the list

```
ls -l molecules/$1 | wc
```

call: `sh count_file.sh ‘*.pdb’`

## SO MANY THINGS TO DO

- The shell does more things than you could ever imagine:
  - sort, cut, regular expressions
  - You can combine as many commands as you want

    -
      ```
      tail -n +6 PATTERNFQ | sed 's:#::g' | grep -v "_wav_rwc" | sed 's:_: :g' | sed
      's:.fits::g' | awk '{if($7==0){print
      $3",PSUB/"$1"_rwc_"$3".fits,"$4","$5","$6","$7} else {print
      $3",FAIL/"$1"_raw_"$3".fits,"$4","$5","$6","$7} }' | sed 's:*:1:g' > tmpinfo.lis
      ```
- Pro: Its fast
- Con: Its usually hard to read