FOAR705 - Learning Journal Week 6 - Software Carpentry Exercises

Jan Jugueta - 44828020

Week 6: 30/8/19 - 6/9/19

30/8/19 - 2:17pm

Going through some Software Carpentry exercises in the FOAR705 class. Some useful commands:

```
Ctrl + C
```

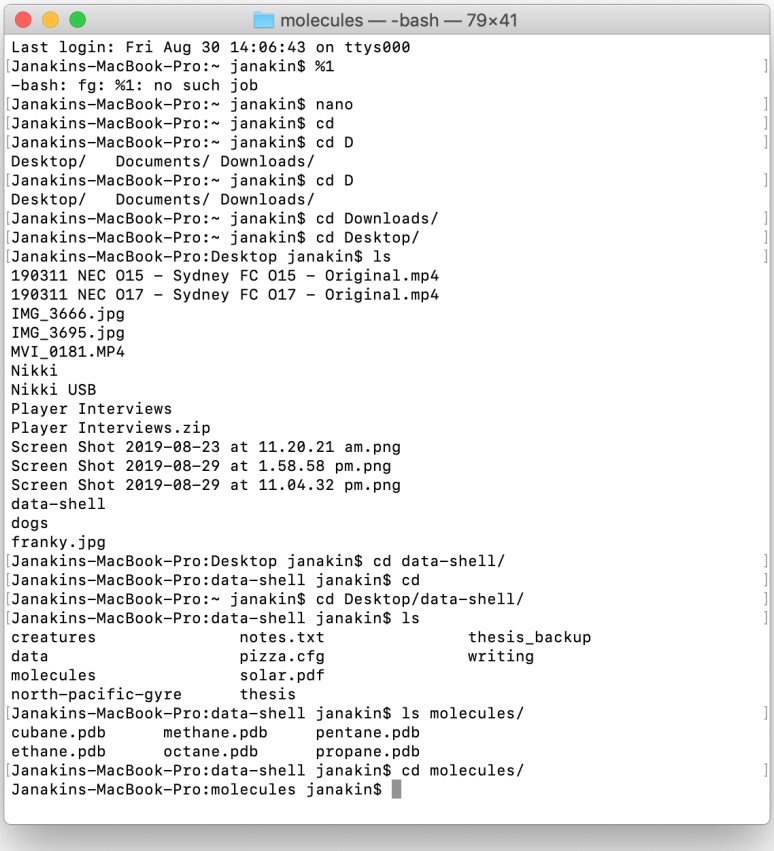
which is cancel.

```
Ctrl + D
```

which is disconnect.

30/8/19 - 2:37pm

Still practicing some Shell things. Practicing how to navigate again.



30/8/19 - 2:43pm

The command

```
history
```

shows all the commands you have used previously.

The command

```
>
```

redirects.

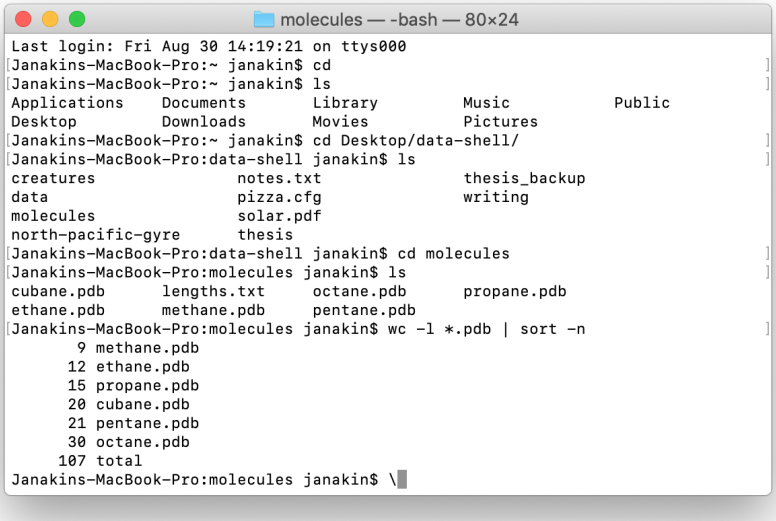
The command

```
cat
```

opens files.

30/8/19 - 2:58pm

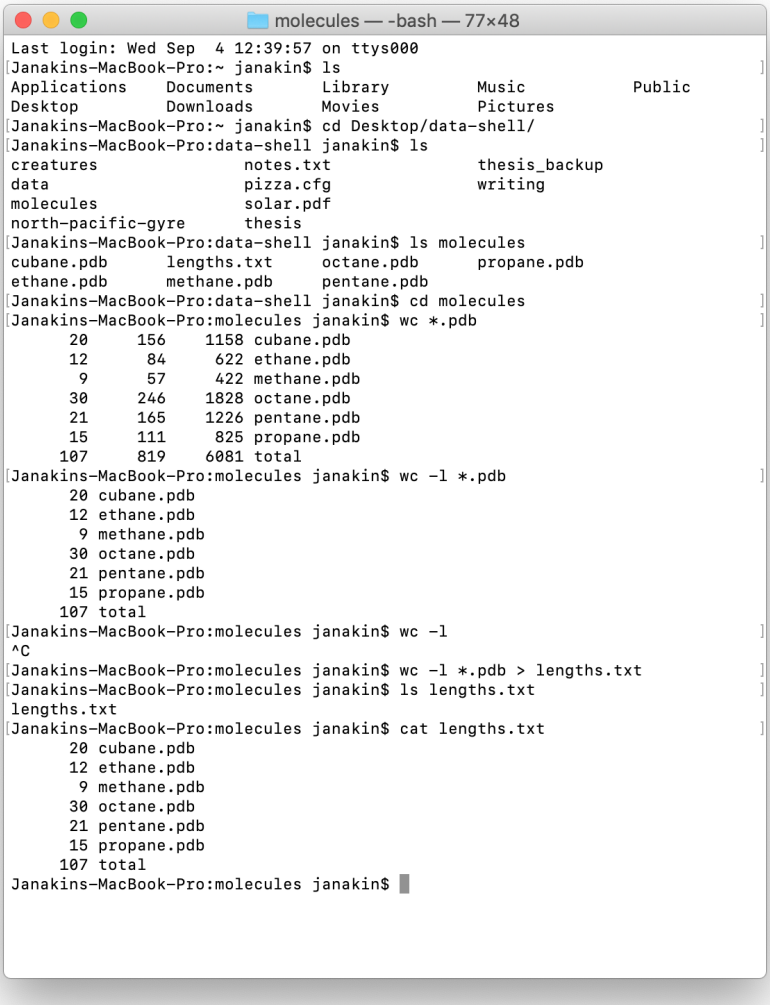
My first pipe attempt.



4/9/19 - 12:38pm

Now time to do the SW Carpentry exercises properly. Continuing with Episode 4 **Pipes and Filters**

Have followed the instructions so far and no errors.

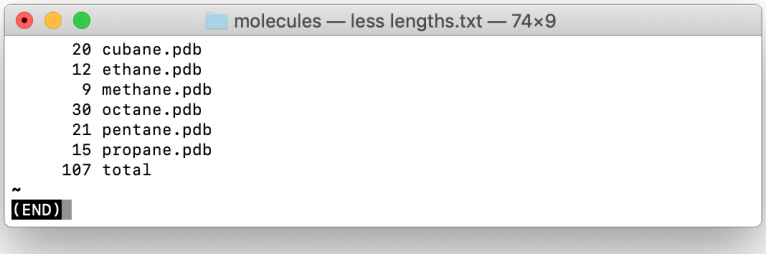


4/9/19 - 12:46pm

The

```
less
```

command in action.



4/9/19 - 12:49pm

The answer to the exercise *What does sort -n Do?* is that

```
sort -n
```

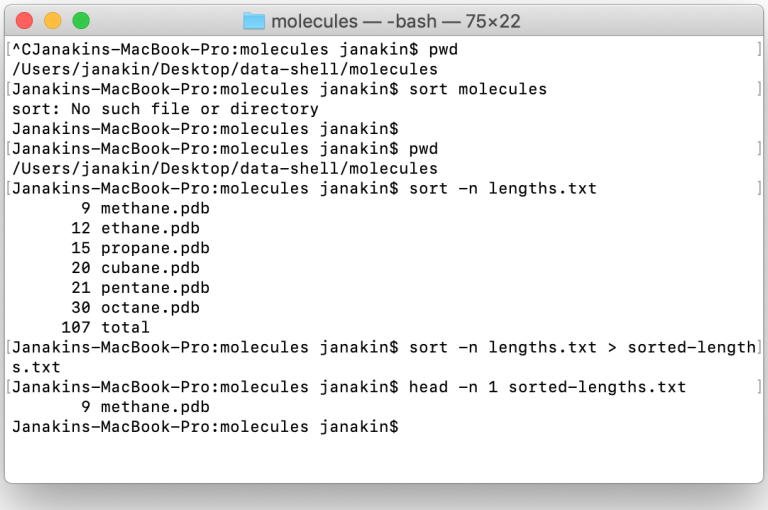
sorts purely numerically, not alphanumerically.

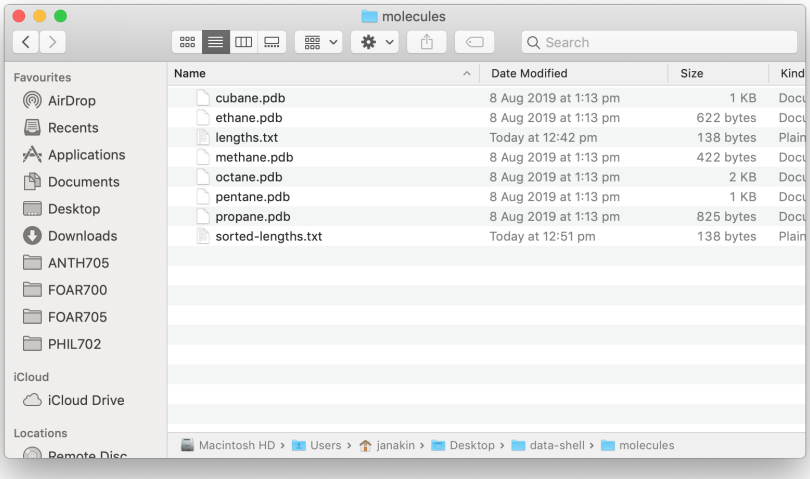
4/9/19 - 12:54pm

Continuing on with the Episode. The following screenshots show the

```
sort -n lengths.txt > sorted-lengths.txt
head -n 1 sorted-lengths.txt
```

commands.





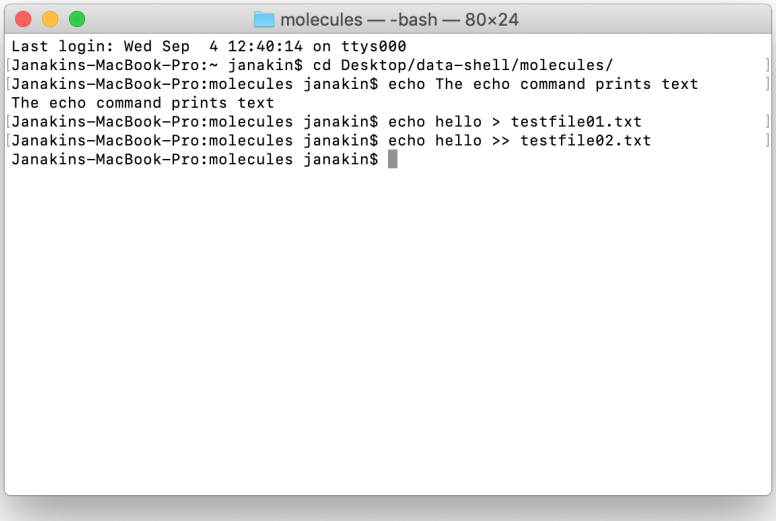
4/9/19 - 12:58pm

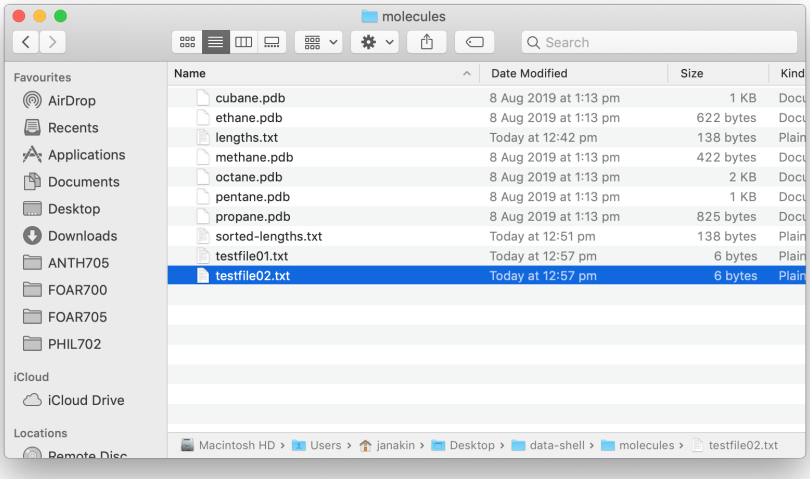
What does

>>

mean?

Well, I've done exactly as the lesson says. But I still don't get it.

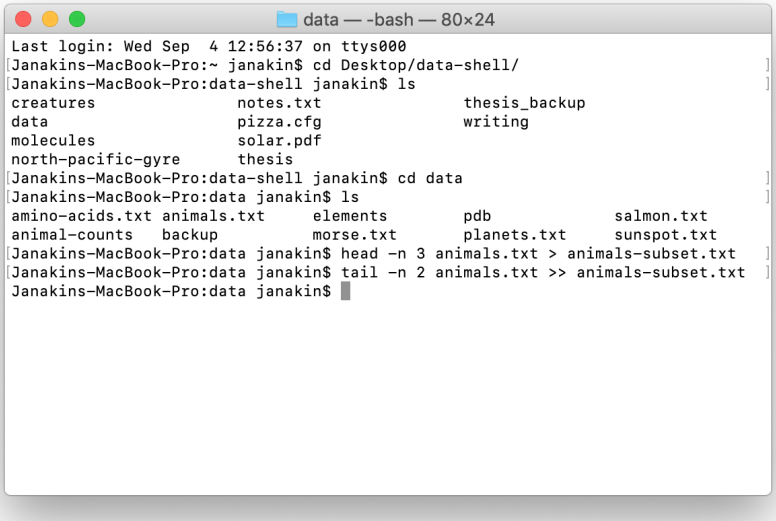


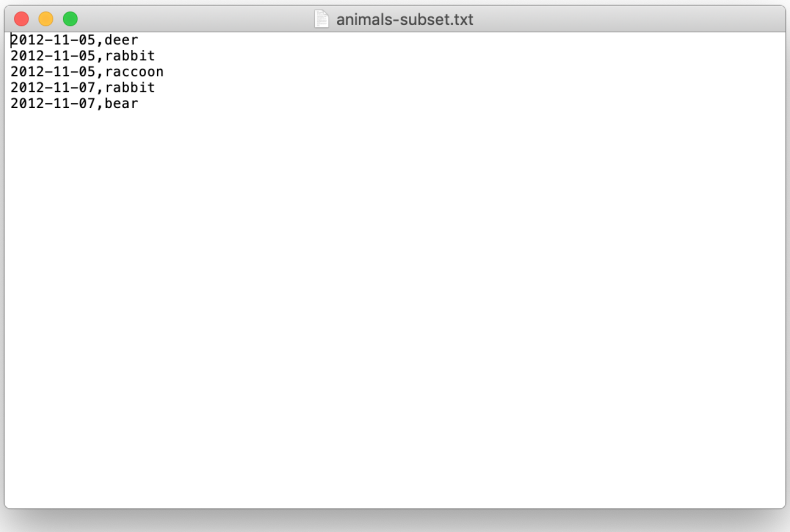


4/9/19 - 12:58pm

The *Appending Data* exercise.

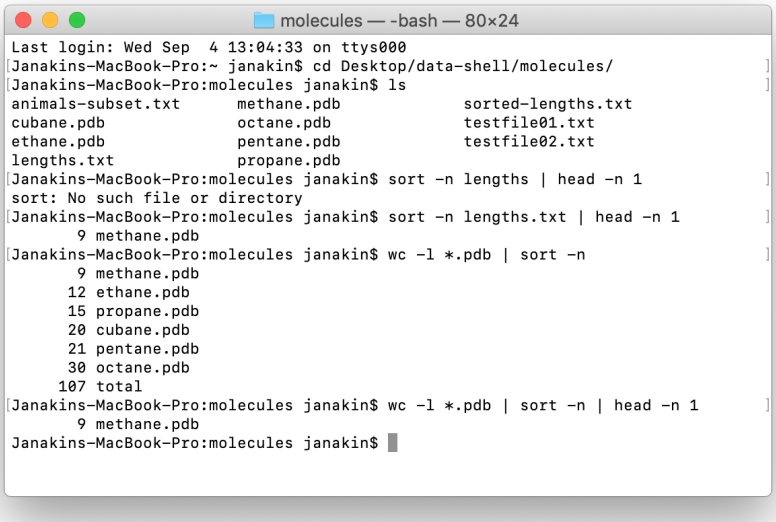
Answer 3. It contains the first three lines and last 2 lines.





4/9/19 - 1:11pm

Having a go at piping.



4/9/19 - 1:13pm

Piping Commands Together exercise.

The answer is option 4.

```
wc -l * | sort -n | head -n 3
```

will get us 3 files which have the least number of lines.

4/9/19 - 1:15pm

Pipe Reading Comprehension exercise.

As expected, I had the same answer as the exercise.

```
2012-11-06,rabbit
2012-11-06,deer
2012-11-05,raccoon
```

4/9/19 - 1:19pm

Pipe Construction exercise.

The solution is

```
cut -d , -f 2 animals.txt | sort | uniq
```

4/9/19 - 1:21pm

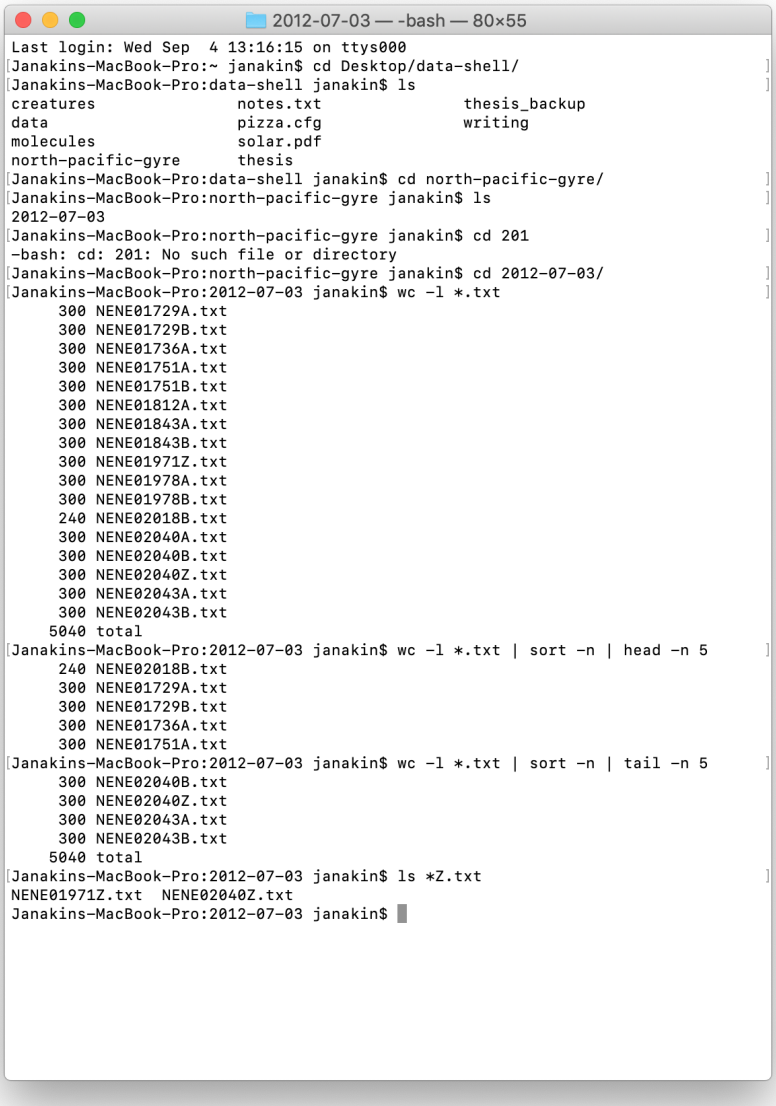
Which Pipe? exercise.

The solution is

```
cut -d, -f 2 animals.txt | sort | uniq -c | wc -l
```

4/9/19 - 1:25pm

Continuing with Nelle's examples.



4/9/19 - 1:27pm

Wildcard Expressions exercise.

Answers.

1.
```
ls *A.txt
ls *B.txt
```

2. They are separated because they are two commands
3. Only when there are no files ending in A.txt or B.txt

4/9/19 - 1:31pm

Removing Unneeded Files exercise.

Only

```
rm *.txt
```

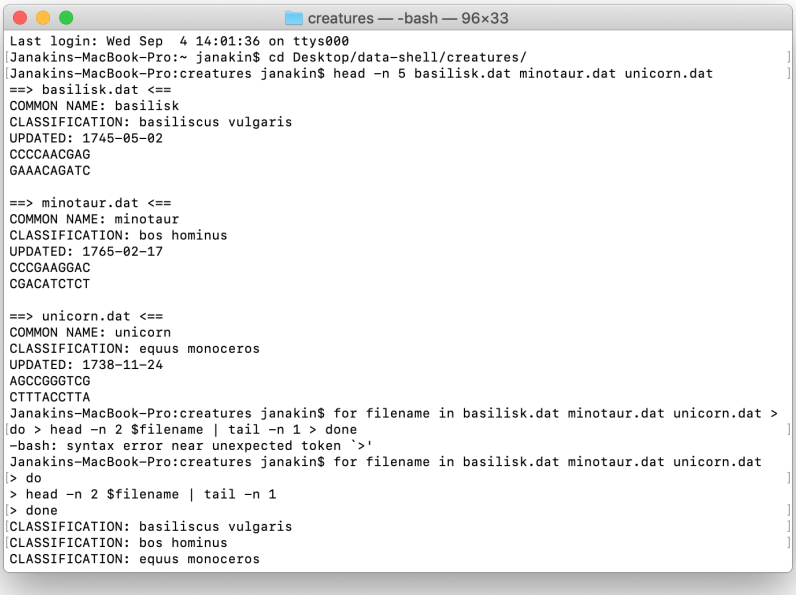
is correct.

4/9/19 - 2:08pm

On to episode 5, **Loops**. Following the instruction, I had realised that I put in the code

```
for filename in basilisk.dat minotaur.dat unicorn.dat
> do
>    head -n 2 $filename | tail -n 1
> done
```

as one continuous code. I then realised that each line was a separate command input.

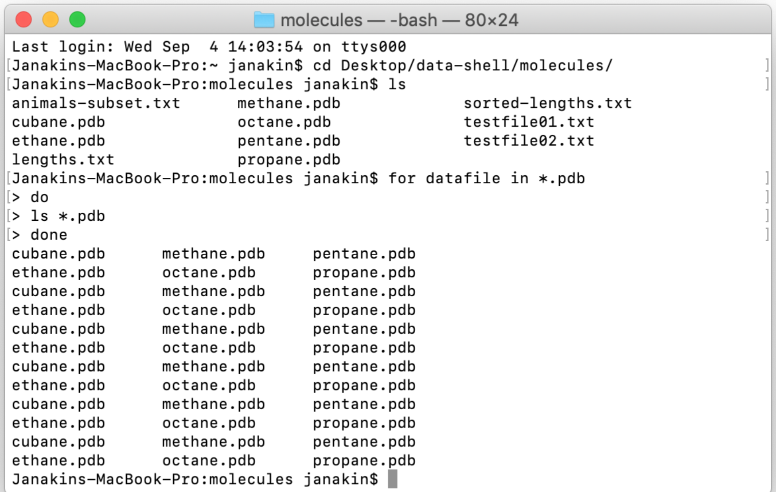


4/9/19 - 2:22pm

Completing *Variables in Loops* exercise. The output of code

```
$ for datafile in *.pdb
> do
>    ls *.pdb
> done
```

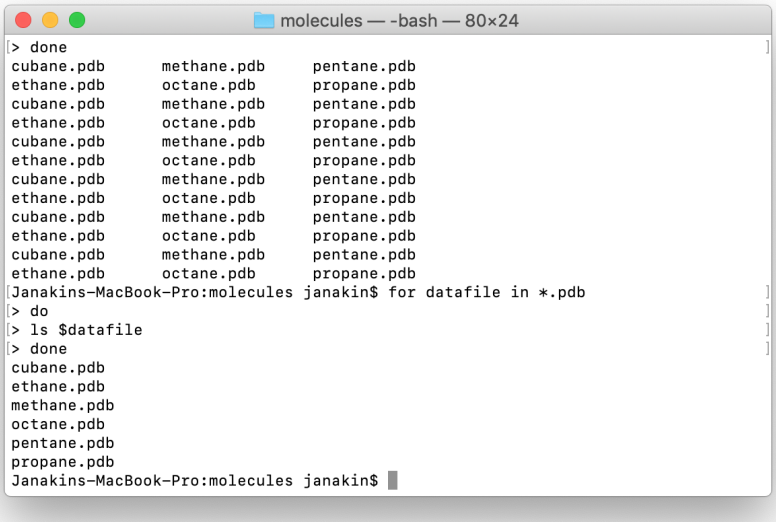
is



The output code of

```
$ for datafile in *.pdb
> do
>ls $datafile
> done
```

is



The second code lists a different file on each loop iteration.

4/9/19 - 2:28pm

Completing *Limiting Sets of Files* exercise.

```
$ for filename in c*
> do
>    ls $filename
> done
```

lists only cubane.pdb.

```
$ for filename in *c*
> do
>    ls $filename
> done
```

lists both cubane.pdb and octane.pdb because of the presence of the c character.

4/9/19 - 2:32pm

Completing *Saving to a File in a Loop - Part One* exercise.

```
for alkanes in *.pdb
do
echo $alkanes
cat $alkanes > alkanes.pdb
done
```

has text from each file written into aklanes.pdb. Each iteration overwrites the last. The propane.pdb is the last to be written in alkanes.pdb.

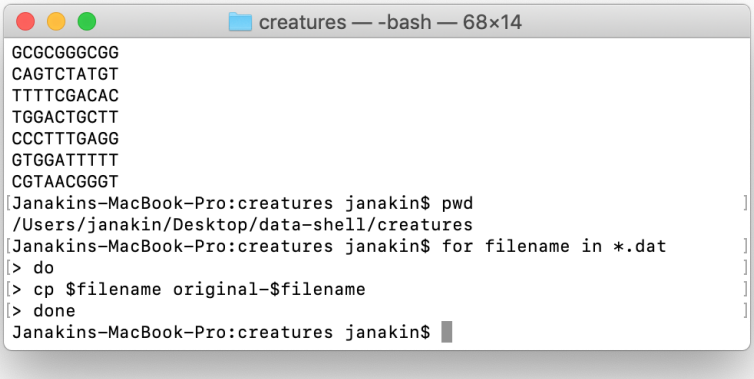
4/9/19 - 2:36pm

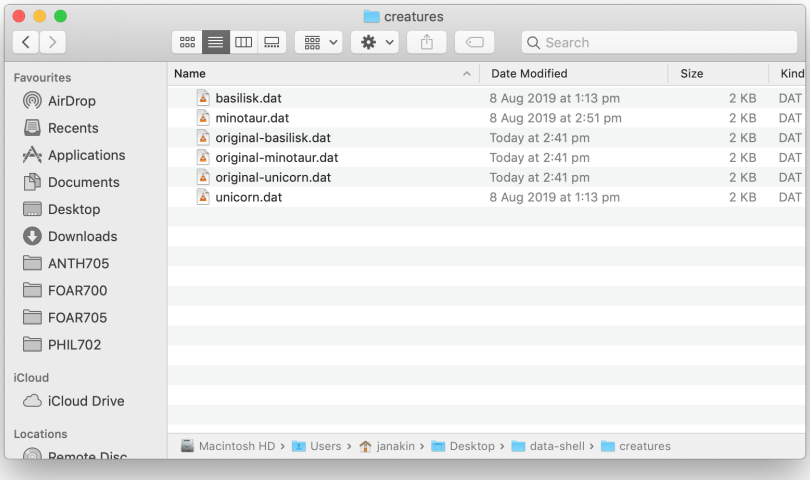
Completing *Saving to a File in a Loop - Part Two* exercise.

All of the text from cubane.pdb, ethane.pdb, methane.pdb, octane.pdb, pentane.pdb and propane.pdb would be concatenated and saved to a file called all.pdb.

4/9/19 - 2:44pm

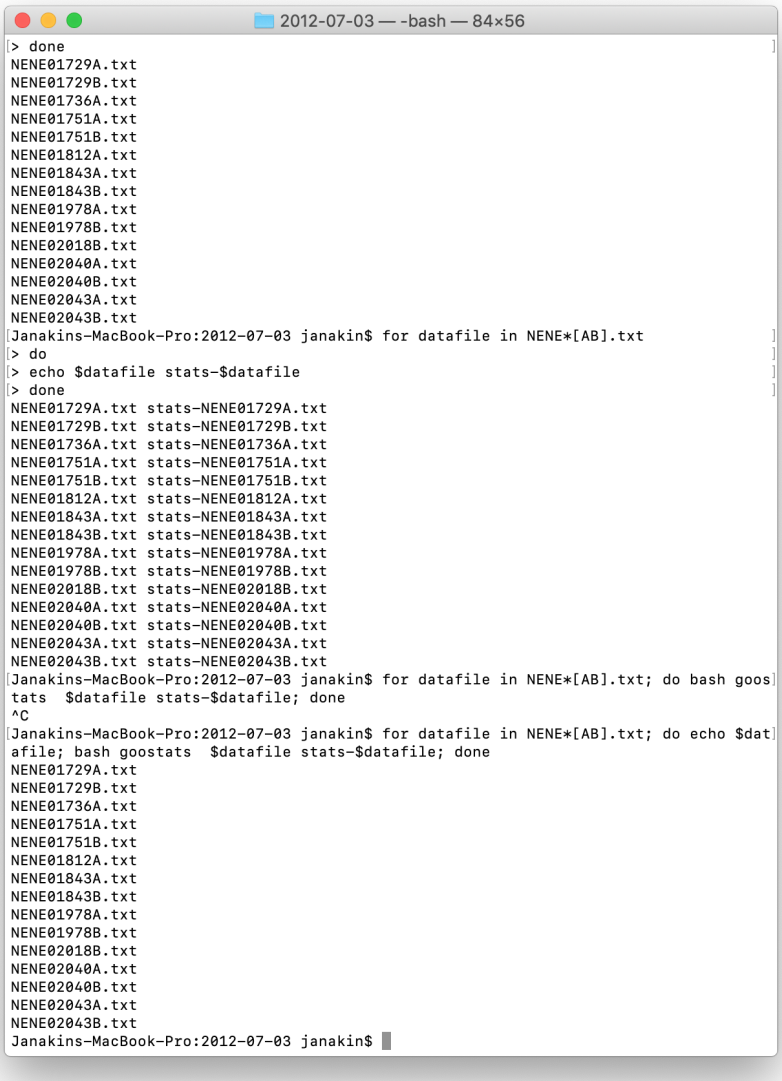
Continuing with the lesson.





4/9/19 - 2:53pm

Back with Nelle's example.



Not exactly sure what bash goostats did, but it did certainly look like Terminal was doing something.

4/9/19 - 2:59pm

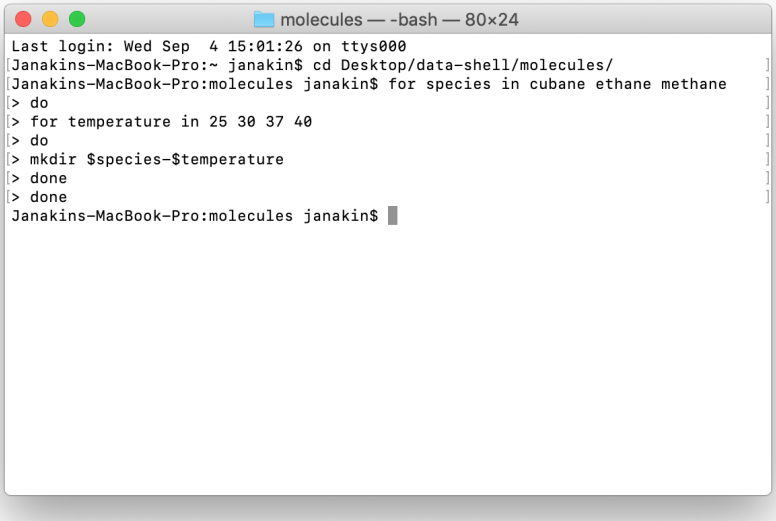
Completing *Doing a Dry Run* exercise.

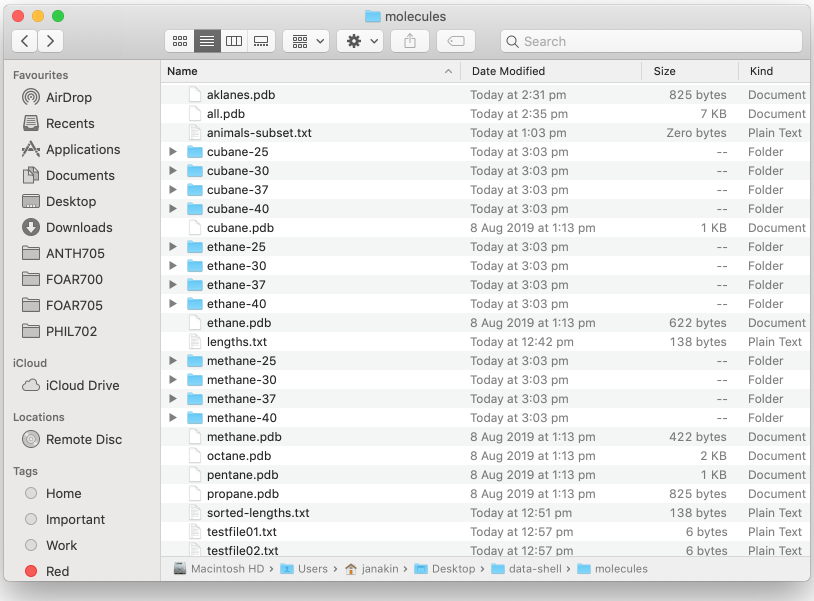
The second version is the one we want to run because it prints everything.

4/9/19 - 3:03pm

Completing *Nested Loops* exercise.

Not sure if this was successful.





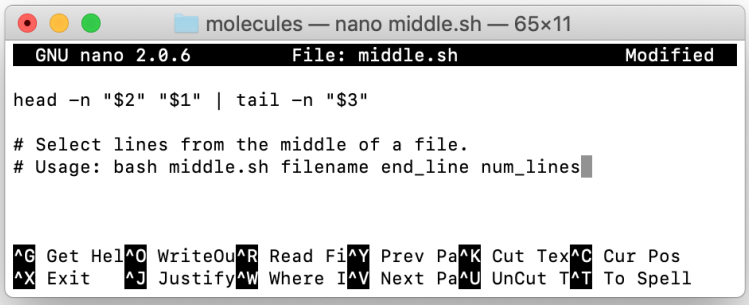
4/9/19 - 3:16pm

Episode 6: Shell Scripts

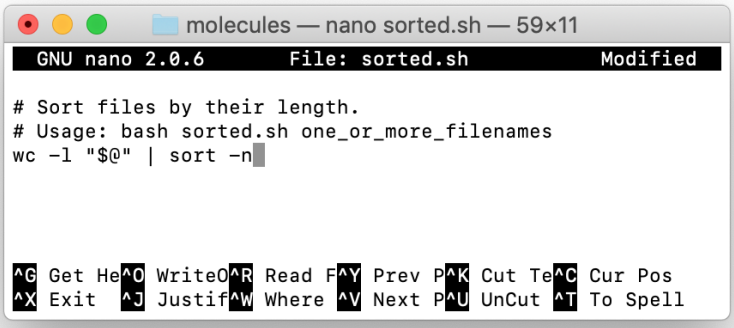
Following the instructions. Below are the screenshots of what I have done.

Creating the middle.sh shell.

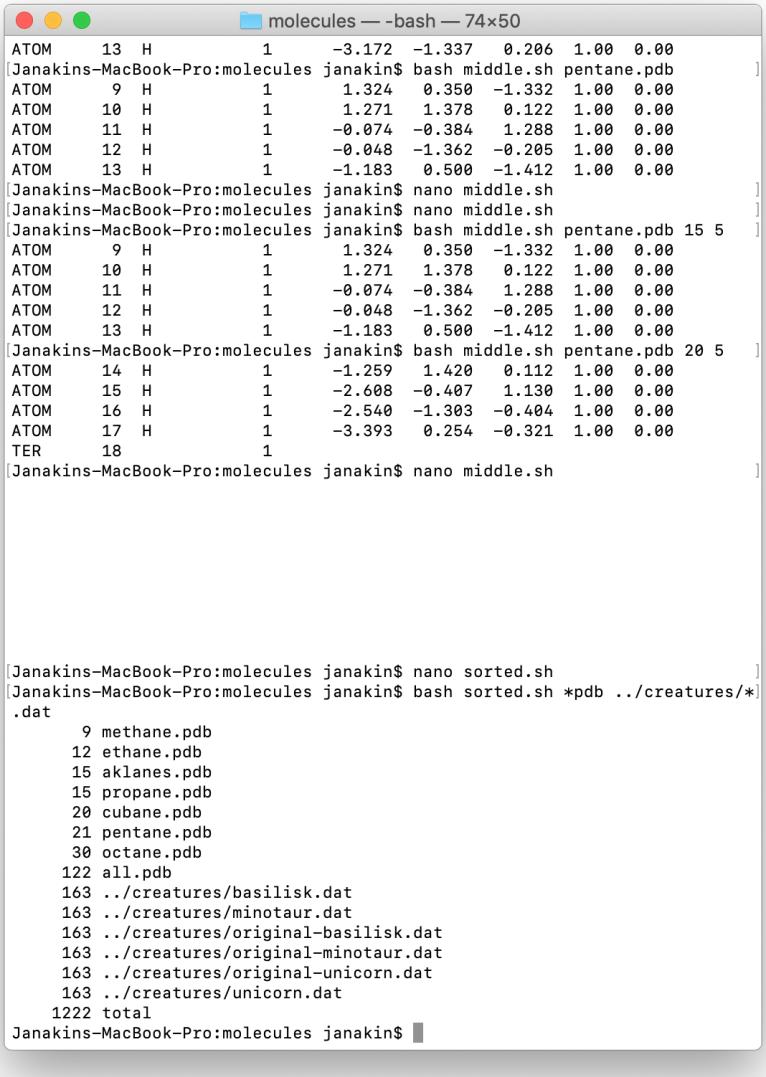
Recoded the middle.sh file to be more versatile and include information to other users about what it does.



Created the sorted.sh shell.



Terminal window and commands used up until this point.



4/9/19 - 3:34pm

List Unique Species exercise. The script I would use.

```
    for file in $@
    do
echo "Unique species in $file:"
# Extract species names
cut -d , -f 2 $file | sort | uniq
    done
```

4/9/19 - 3:36pm

Why Record Commands in the History Before Running Them

It is a good idea to do this because it gives you a record of what commands you used, and in what order. So if things go wrong, you can trouble shoot.

4/9/19 - 3:41pm

Following on with Nelle's example.

4/9/19 - 3:43pm

Variables in Shell Scripts

The correct answer is 2.The first and the last line of each file ending in .pdb in the molecules directory

4/9/19 - 3:44pm

Find the Longest File With a Given Extension

I would use the code

```
wc -l $1/*.$2 | sort -n | tail -n 2 | head -n 1
```

4/9/19 - 3:47pm

Script Reading Comprehension

Script 1 would print out a list of all files containing a dot. Script 2 would print the contents of the first 3 files with a .pdb extension. Script 3 would print all the arguments to the script followed by .pdb.

4/9/19 - 3:48pm

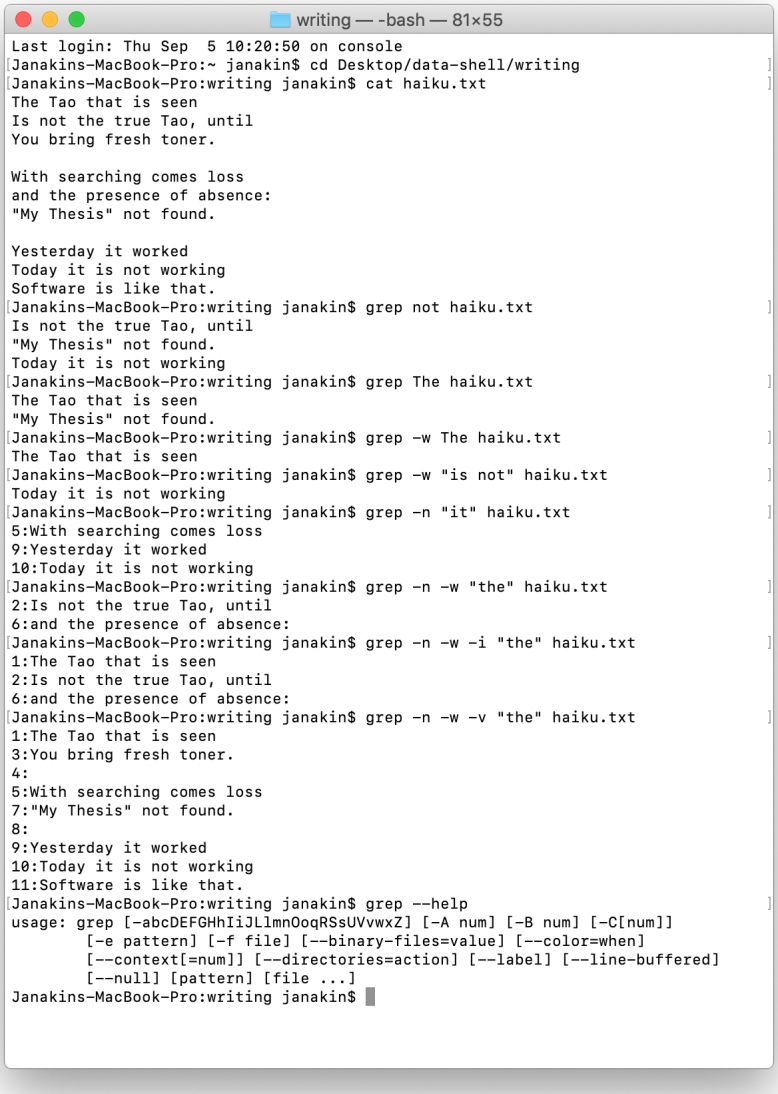
Debugging Script

The -x option gets bash to run in debug mode, printing out each command as it runs it, which will help when trouble shooting.

4/9/19 - 10:44am

Am now on the 6th and final episode **Finding Things**.

The first little bit.



4/9/19 - 10:50am

Using grep

The correct answer is 3

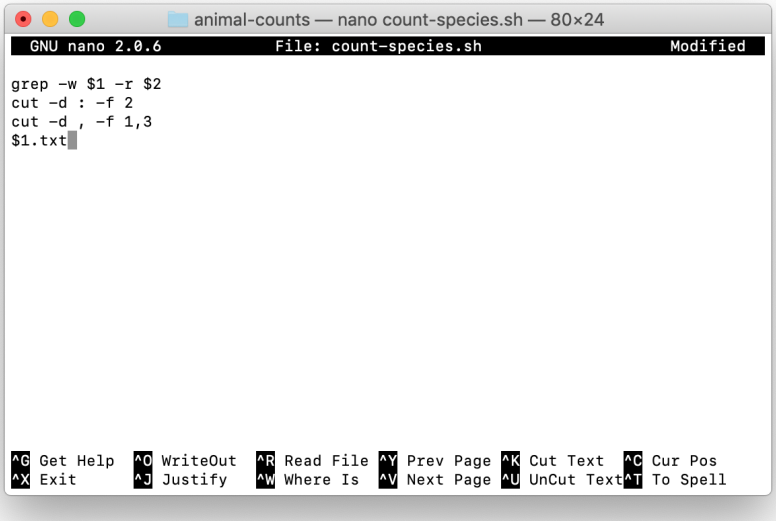
```
grep -w "of" haiku.txt
```

because it will only look for whole words.

4/9/19 - 10:57am

Tracking a Species

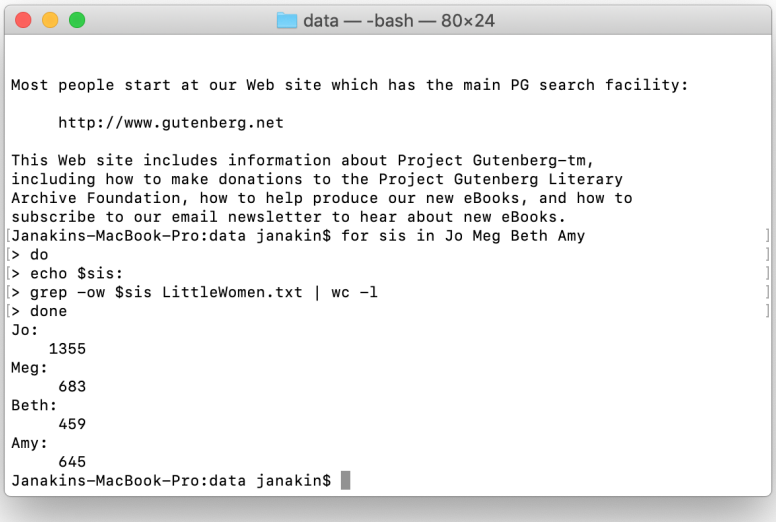
I made this shell script.



4/9/19 - 11:00am

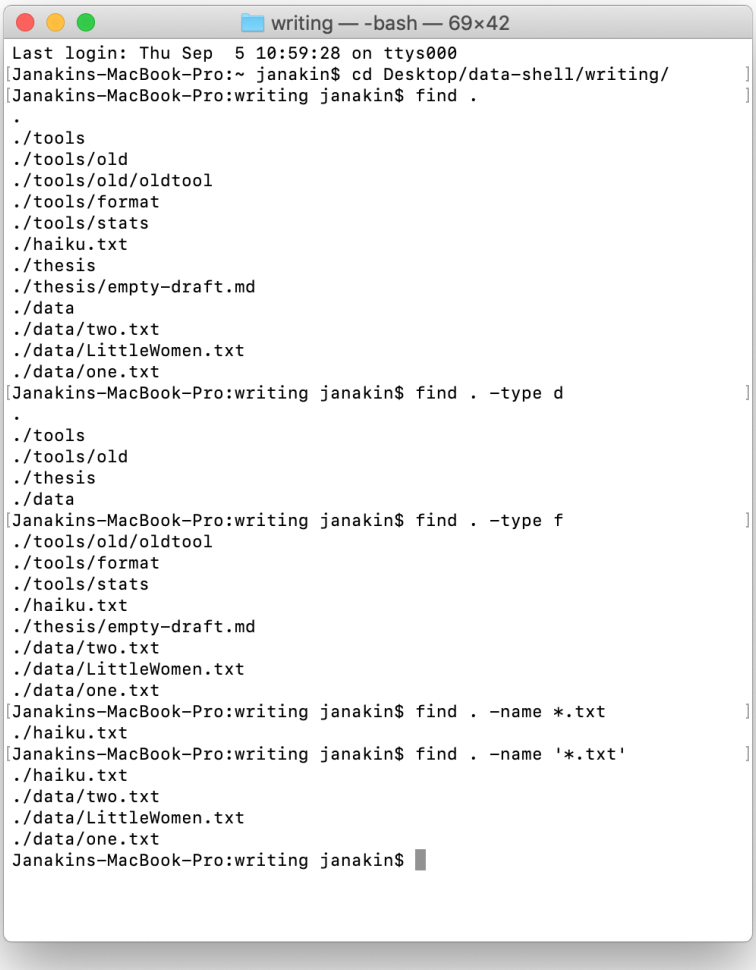
Little Women

I did the following.



4/9/19 - 11:04am

Continuing on with the lesson.



4/9/19 - 11:06am

Matching and Subtracting

The correct answer is 1.

```
find data -name '*s.txt' | grep -v net
```

4/9/19 - 11:10am

find Pipeline Reading Comprehension

It will find .dat files, count the number of lines the files have and sort the output numerically.

4/9/19 - 11:12am

Finding Files With Different Properties

```
find ./ -type f -mtime -1 -user ahmed
```