WRITE ON BOARD: username and password info for logging into AWS VM

Welcome! Linux shell workshop. If you know what files and directories/folders are and you can type, you're qualified to be here. This is a beginner workshop.

Introductions
Linux shell: Why are you here? What do you want to get out of this workshop?

What is the shell?

- Back in the day, 1970s, 80s into 90s main way that you used a computer was to type in commands, one at a time. This is called the command line.
- You could also write programs, run them from the command line.
- You ran commands in what is called the shell-- it wraps around the operating system, giving you a set of consistent commands you could use to interact with the computer and its files.
- The operating system, UNIX, and then Linux, became really popular and because the shell commands were so useful and powerful, they're widely used today. You can use them on Macs, Linux computers, and other networked computers or servers you might use if you're working with a lot of data or software. (bash)

What's so great about working in the shell?

- Versatile commands, great for managing lots of files, exploring files, working with data in lots of different formats. String together commands.
- You can automate tasks for working with files.
- Lots of other packages you can install to add to what you can do on the command line.

Downsides:

Commands can be cryptic, shortened words and abbreviations.
Computer: type everything exactly right. Spaces, hyphens, dots, punctuation…

Lots of different computers here today:
Mac, you have Terminal application: it lets you use Linux commands on the files on your computer.
Windows Doesn't natively support using Linux commands, it's has its own set of command line tools, in the windows OS NOT Linux shell commands.

So we are all going to connect to a server that has Linux. This is also something you might need to know how to do anyway if you're working with data on another server.

To access that Linux computer, we're going to use an application that lets us connect in a secure way, and gives us a shell where we can run commands.

COUNT OFF TO CREATE USERNAMES AND PASSWORDS

Macs: Terminal Windows: PuTTY …. USE testuser3

GOAL 1: MOVING AROUND AND COPYING FILES

Flower Data

Find out where you are: (all of you are in different places, different home directories)

```
pwd
```

List the contents of the current directory, use an option, find out about all of them

```
ls
ls -l (options for commands!)
man ls
```

Look there's a directory

```
cd ..
pwd
cd flowerdata (argument)
```

Let's make backup copies of all of these files.

```
cp iris.csv iris.csv.bak (multiple arguments)
cp iris1.csv iris1.csv.bak
cp iris2.csv iris2.csv.bak
```

Make a new directory called 'backups'

```
mkdir backups
ls
```

Copy the flower data files from to the backups folder. Hint, to save time use the wildcard operator: *

```
cp *.bak backups/
```

Move to the new directory you just created

```
cd backups
```

relative vs. absolute paths

Print out the contents of the files, look at just the first few lines or the last few lines.
cat / head / tail - look at the files without editing them.

```
more iris.csv
cat iris.csv
head iris.csv
head -20 iris.csv
head -n 20 iris.csv
tail iris.csv
```

These are csv files, and we'd like to merge them into one file. Delete the header rows on the second two files so they can be merged cleanly. Different ways to do things...

```
nano iris2.csv
tail -n +2 iris3.csv
tail -n +2 iris3.csv > iris3data.csv
tail -n +2 iris3.csv >> iris.csv
```

Littlewomen novel text

```
mkdir littlewomen
```

```
cd littlewomen
```
(learn tab completion)

```
gunzip littlewomen.txt.gz
```

```
wc on the file
```

```
grep Amy searchdata.txt
grep Amy *
grep Amy | wc -l

grep Amy * | more
grep 'and Amy' *
```

grep not | wc -l
grep -w not | wc -l (has to be a whole word)
cat | sort (or sort -f -- show use of sort --help)
cd ..

Search data

1. `cd searchdata`
2. `gunzip searchdata.txt.gz`
3. `wc -l searchdata`
4. `sort searchdata | uniq | wc -l`
5. `sort searchdata | uniq > uniquesearches.txt`
6. `history`
7. `history | more`
8. `history | grep mkdir`

Cowsay

9. Try installing w/o sudo
10. Install w/sudo
11. Try again, it works

12. (use quotation marks)
13. install fortune
14. fortune | cowsay
15. man fortune
16. shuf -n 1 searchdata.txt | cowsay

Other stuff

17. man wget
18. man sort
19. sort --help
20. man ls

If there's extra time:

- group with (), literal with quotes - for example file names with spaces in them
- `command` - sends output to stdout
- time *anothercommand*
- top
- du / df
- sort -k 3 -- sorts based on the 3rd column of data
- sort -n -- sorts numerically
- find . -name myfile
- find . -print
- file names in UNIX can have spaces in them. Use quotes when referencing a file name w/spaces (e.g. cat my file.txt)

SCRIPTING:

- create a file like:

```
#!/bin/bash
wc * | sort -k 3 -n | head -n 1
```

- try running the file - no permissions
- chmod a+x the file
- try running now

https://github.com/jlevy/the-art-of-command-line/blob/master/README.md