

# How to Babbage

Here we are going to try and work through some basics of the terminal and some tasks you might need to do.

# Stuff you see that looks like this:

```
codey stuff  
more codey stuff
```

Means you enter it into the terminal, or that's what comes out

# Baby steps

## ***WHERE AM I!***

`pwd` = print working directory

```
pwd  
/c/Users/Seth/Documents/GitHub
```

# Where do I go?

`cd somedirectory` `cd` = change directory

`cd ..` : up a directory

# Make friends with TAB

Start typing while you are changing directory and hit **TAB** to have the terminal try and autocomplete

# Make a testFile or three

Make an empty file with not-at-all-creepyly-named command

```
touch testFile.txt  
touch testFile2.txt  
touch testFile.tab
```

## Put something in the file

```
echo 'testeyteddyeyesty' >> testFile.txt  
echo 'blahblahblah' >> testFile.txt  
  
echo 'plerplerggggkl;jkljkl;' > testFile2.txt  
echo 'nimbynimby' > testFile2.txt
```

## How did that work

```
head testFile.txt
```

```
head testFile2.txt
```

`>` is not the same as `>>`

`>` sends it to the file and replaces the whole file.

`>>` appends to the file (does not replace the file)

`cat` catenates 🐱

```
cat testFile.txt testFile2.txt > comboFile.txt
```



# Looking at stuff

`head` shows the top of a file

`tail` is the bottom (surprise!).

`less` shows you more 🙄 (type `q` to exit)

What's here? list = ls

```
ls
```

ooh, that's kinda ugly

try `ls -l` (-l is the 'flag' for long)

try `ls -al` (... -a all)

try `ls -alh` (... -h human readable)

try `ls -alht` (... -t ordered by time)

try `ls *.txt` (just the .txt files)

# Moving and ~~shaking~~ copying

OK, you made a mistake and want to rename something

```
mv oldname newname
```

`mv` = move, eg:

```
mv testFile.tab testTab.tab  
ls
```

# Moving and ~~shaking~~ copying

Let's be safe and copy (`cp`) a file we want to mess around with

```
cp testTab.tab backupTab.tab  
ls
```

# Removing files



Warning: terminal will do what you tell it. Even if that means delete everything. Proceed with caution and when in doubt test first. eg to come

`rm` = remove

```
rm backupTab.tab
```

You can use wildcards (eg. \*.tab would be all .tab files) but it's dangerous. Dont' go throwing wildcards around until you are confident

If you want to use wildcards with deleting (rm) test first eg:

`ls *.tab` does that list only the files you want to remove? If so, go ahead and replace `ls` with `rm`. If not, stop!

# Connect to the server

```
ssh barribeau@138.253.198.78
```

## Tips and tricks

Open hidden file for ssh configuration in nano (text editor)

```
nano ~/.ssh/config
```

add the following (but with your deets obv):

```
Host babbage
    HostName 138.253.198.78
    User barribeau

Host euler
    HostName euler01.liv.ac.uk
    User seth
```

exit by typing `cntrl+x` and saying yes you'll save ( `y enter` )



Now you can connect with easier to remember:

```
ssh babbage
```

**Tadaaaah** 

# Getting stuff onto the server

Transfer with `scp`

with the shortcut

```
scp testFile.txt babbage:path/to/where/you/want/it/to/go
```

without the shortcut

```
scp testFile.txt barribreau@138.253.201.241:path/to/where/you/want/it/to/go
```

# Getting stuff off the server

Transfer with `scp`

with the shortcut

```
scp babbage:path/to/where/your/file/is/testFile.txt path/to/put/it/on/your/computer
```

without the shortcut

```
scp barribeau@138.253.201.241:path/to/where/your/file/is/testFile.txt path/to/put/it/on/your/computer
```

# Stuff that takes a long time

Use a `screen` . Screens allow you to start something and run it in the background. You can then connect to it later. This is very useful because

- Sometimes connections suck and you get cut off.
- You may need to use your computer for other stuff.
- Life happens
- Use `screen` it will save you tears and years.

make a new screen

```
screen -S myNewScreen
```

you now have a new screen running and you start your things.

To exit the screen you type `cntrl+a` then `cntrl+d` and you will be out of the screen.

To reconnect you just type

```
screen -r myNewScreen
```

When you are done, type `exit` to finish that screen.

# Downloading stuff

`wget` or `curl` will download files.

e.g. you want to get [this data](#) from the SRA. You then click through to the [RUN ID](#) and see the download link under data access.

```
wget https://sra-download.ncbi.nlm.nih.gov/traces/sra0/SRR/012234/SRR12527928
```

# Oh no, that's going to take for ever!

## Cancel!

`cntrl+c`  you may use this a lot

get rid of that partial file

```
rm SRR12527928
```

# Decompressing stuff

Downloaded files are usually compressed in one way or another. Which means you have to extract them. Which I always forget how to do. Remember your Google-fu

if it's `*.zip` then, you `gunzip randoFile.zip`

If it's some flavour of `tgz/tar.gz/etc...` Look it up but it will look a bit like this

```
tar -zxvf randoFile.tar.gz
```



It may seem scary, but you've got this

