Command Line for Creatives

Orientation: Who are we and where are we?

Explanation	Command
Print "hello world" to the screen	`echo "hello world"`
Print the name of the current user account	`whoami`
Print working directory (directory=a folder)	`pwd`
Make a directory called files	`mkdir files`
List the contents of the current directory	`ls`
List the contents of the current directory, including hidden files and their metadata	`ls -la`
Change directory to files (enter the "files" directory you just made)	`cd files`
Go back up a directory (to where you just were)	`cd ..`
Change directory to files	`cd files`
Create a file called file.txt (or update the timestamp of file.txt if it already exists)	`touch file.txt`
Clear the terminal	`clear`

Shortcuts

These may not all work on your OS or shell, but in general:

```
Tab --> autocomplete
Up Arrow --> Recall previous commands
Ctrl-A --> Jump to start of line
Ctrl-E --> Jump to end of line
Ctrl-K --> Cut a line
Ctrl-Y --> Paste a line
Ctrl-C --> Exit a running command/process
```

On Linux:

```
Ctrl+Shift+T --> Open a new terminal
Ctrl+Shift+C --> Copy
Ctrl+Shift+V --> Paste
```

File Manipulation

Explanation	**Command**
Create a copy of file.txt and call it file2.txt	`cp file.txt file2.txt`
Create a copy of file2.txt and call it file3.txt	`cp file2.txt file3.txt`
Use `ls` to see that the files were created as you expected	`ls`
Remove file.txt	`rm file.txt`
Move (and rename) file2.txt to file.txt	`mv file2.txt file.txt`
Use `ls` to see that the files were created as you expected	`ls`
Make a directory called docs	`mkdir docs`
Move file.txt into docs	`mv file.txt docs/`
Move all .txt files into docs (* is a wildcard operator)	`mv *.txt docs/`
Move all files that start with "file" into docs	`mv file.* docs/`
Use `ls` to see that the files were created as you expected	`ls`
List the contents of docs	`ls docs`
Remove the docs folder and all it's contents (be careful with this!)	`rm -rf`

Redirecting

Explanation	Command
Create a file called new_file.txt	`touch new_file.txt`
Write the words "Hello World" to new_file.txt	`echo "Hello World" > new_file.txt`
Print the contents of new_file.txt to the terminal (stands for concatenate)	`cat new_file.txt`
Append "How are u" to the end of new_file.txt	`echo "How are u" >> new_file.txt`
Check what it contains now	`cat new_file.txt`

Note that the `>` operator will replace the contents of the file with the new contents, and the `>>` operator will append to the end.

Many commands can be piped into one another using the `|` pipe operator. Some examples:

Explanation	Command
Pipe echo to a tool for counting words	`echo "how many words do i have" | wc -w`
Pipe a list of all running processes to a tool for searching for a specific string	`ps aux | grep "python"`
Pipe a web request for this document to a tool for searching	`curl "https://hackmd.io/rKCLMlAzRoikoRNEmdntKw" | grep "docs"`

Break

Connecting to Other People's Computers

Connect to a server at `142.93.189.250` using a tool called `ssh` (stands for Secure Shell) as a given user, then verify your username and current working directory:

```
ssh user@142.93.189.250
whoami
pwd
```

Note we set up this server for the workshop and it is no longer running, but leaving this block as an example/reference

Running a Program

You can run many programs from the shell just by typing their name, for example, `nano` is a text editor that comes installed on many operating systems. (*Note if you're using windows it's not installed by default*)

```
nano
```

In order to exit nano, you can use `Ctrl+x` then `y`. (*Note if you've added anything to the file, you will then be promped for a filename also*). You can check the location of the program using `which`:

```
which nano
```

Something else installed by default might be python.

```
python
```

You can exit python using `quit()`. If python is not found, you might have `python3` instead. You can check it's location on your system using `which python` or `which python3`.

The way your computer finds these programs is using the `PATH`. The path is built into your shell as an environment variable, and it contains a list of places for the shell to look when you ask it to run a command/program. You can check what's on it using:

```
echo $PATH
```

If a program is in a location not on the path, you will need to run it from the location it's stored in, reference it by it's location (ie. `/home/user/Documents/my_program/`) or modify the path.

Scripting

Use nano to open (or create) a file called file.txt

```
nano my_program.sh
```

Add this to the file, use `Ctrl+x` then `y` to save and quit.

```
#!/bin/sh
echo "Hello World"
```

The first line is called a shebang, and is a magic piece of code your computer uses to recognize that this is a bash script. The second line should be familiar.

You now need to make your file executable, which will allow the computer to run it directly as a program, instead of only reading and writing to it like a normal file. You may remember when we did `ls -la` earlier it printed a string of characters looking something like `-rw-rw-r--`. The `r` 's and `w` 's stand for read and write.

```
chmod +x my_program.sh
```

After running, if you do `ls -la` you will see the file permissions are `-rwxrwxr-x`, including `x` for executable.

You're now ready to run the program!

```
./my_program.sh
```

It should print "Hello World" to the terminal

Save your script to your own computer!

Sftp (secure file transfer protocol) is a tool like ssh that allows you to easily transfer files. Some of the folks who stayed late after class gave it a try. It needs to be run in a new terminal and would connect to the same server we used during class. This is an example of how to download the script file to your own computer:

```
sftp user@142.93.189.250
get my_program.sh
```

Note - again - we set up this server for the workshop and it is no longer running, but leaving this block as an example/reference

:･ﾟ✧:･ﾟ✧ Fun and Games :･ﾟ✧:･ﾟ✧

Some useful or fun command line programs you may already have installed (or want to try):

```
bastet
lynx
cowsay
traceroute
telnet
curl
```