Introduction to Python

Goals

At the end of this workshop:

1. Students will be able to demonstrate basic navigation in a bash terminal.
2. Students will be able to demonstrate understanding of python syntax.
3. Students will be familiar with Jupyter Notebooks and its usage in data science.
4. Students will be familiar with File I/O (input output) operations.
5. Students will demonstrate understanding of python modules and libraries.

fournova

COMMAND LINE CHEAT SHEET

presented by Tower - the best Git client for Mac and Windows

DIRECTORIES

```
$ pwd
```
Display path of current working directory

```
$ cd <directory>
```
Change directory to <directory>

```
$ cd ..
```
Navigate to parent directory

```
$ ls
```
List directory contents

```
$ ls -la
```
List detailed directory contents, including hidden files

```
$ mkdir <directory>
```
Create new directory named <directory>

OUTPUT

```
$ cat <file>
```
Output the contents of <file>

```
$ less <file>
```
Output the contents of <file> using the less command (which supports pagination etc.)

```
$ head <file>
```
Output the first 10 lines of <file>

```
$ <cmd> > <file>
```
Direct the output of <cmd> into <file>

```
$ <cmd> >> <file>
```
Append the output of <cmd> to <file>

```
$ <cmd1> | <cmd2>
```
Direct the output of <cmd1> to <cmd2>

```
$ clear
```
Clear the command line window

FILES

```
$ rm <file>
```
Delete <file>

```
$ rm -r <directory>
```
Delete <directory>

```
$ rm -f <file>
```
Force-delete <file> (add -r to force-delete a directory)

```
$ mv <file-old> <file-new>
```
Rename <file-old> to <file-new>

```
$ mv <file> <directory>
```
Move <file> to <directory> (possibly overwriting an existing file)

```
$ cp <file> <directory>
```
Copy <file> to <directory> (possibly overwriting an existing file)

```
$ cp -r <directory1>
  <directory2>
```
Copy <directory1> and its contents to <directory2> (possibly overwriting files in an existing directory)

```
$ touch <file>
```
Update file access & modification time (and create <file> if it doesn't exist)

PERMISSIONS

```
$ chmod 755 <file>
```
Change permissions of <file> to 755

```
$ chmod -R 600 <directory>
```
Change permissions of <directory> (and its contents) to 600

```
$ chown <user>:<group> <file>
```
Change ownership of <file> to <user> and <group> (add -R to include a directory's contents)

SEARCH

```
$ find <dir> -name "<file>"
```
Find all files named <file> inside <dir> (use wildcards [*] to search for parts of filenames, e.g. "file.*")

```
$ grep "<text>" <file>
```
Output all occurrences of <text> inside <file> (add -i for case-insensitivity)

```
$ grep -rl "<text>" <dir>
```
Search for all files containing <text> inside <dir>

NETWORK

```
$ ping <host>
```
Ping <host> and display status

```
$ whois <domain>
```
Output whois information for <domain>

```
$ curl -O <url/to/file>
```
Download <file> (via HTTP[S] or FTP)

```
$ ssh <username>@<host>
```
Establish an SSH connection to <host> with user <username>

```
$ scp <file>
  <user>@<host>:/remote/path
```
Copy <file> to a remote <host>

PROCESSES

```
$ ps ax
```
Output currently running processes

```
$ top
```
Display live information about currently running processes

```
$ kill <pid>
```
Quit process with ID <pid>

30-day free trial available at
www.git-tower.com

TOWER
The best Git Client for Mac & Windows

Common Terminal Commands (15 mins)

ls - lists all the files in the directory

pwd - print working directory

cd <dir location > - change working directory

1. cd .. or cd../
2. cd ~

mkdir <name of directory> - make directory

touch <filename> - create a file in the directory you are in

cat <filename> - output file contents

echo <filename> - write arguments to the standard output.

rm <filename> - remove file

mv <filename> <filename> - move file

cp <filename> <filename>- copy file

Exercise: (Checkpoint)

1) Setup your own MDST directory only in the terminal by:
   a) Traversing root->documents->mdst_workshops
   b) Making three directories:
      i) Workshop1
      ii) Workshop2
      iii) Workshop3
      iv) Exercises
   c) Creating a file in workshop1
   d) Copying that file to workshop2
   e) Creating a new file in workshop2
   f) Moving the file to workshop3
   g) Printing out the file content
   h) Adding content to the file

[This would give practice and is supposed to cover all the commands above to some degree. They can push this to GitHub as well]

Python: Introduction (1.5 hours)

Exercise 1:

Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. **Hint: how does an even / odd number react differently when divided by 2?**

Extras:

1. If the number is a multiple of 4, print out a different message.
2. Ask the user for two numbers: one number to check (call it num) and one number to divide by (check). If check divides evenly into num, tell that to the user. If not, print a different appropriate message.

Exercise 2: (HW)

Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right. (Hint: remember to use the user input lessons from the very first exercise). Keep the game going until the user types “exit”. [try checking the random module in python on google. Concepts: Infinite loops, if, else, loops and user/input].

Exercise 3: (HW, harder , Simple Interview Question)

Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)

Concepts: List/String indexing

[Hint: for i in range(len(word)) ; and len(word) - i - 1] will give you the word from the back.

Python: Libraries and File I/O (15 mins)

Exercise:

Create a function to read name and password of “secret.txt” which you have created.

a. Use the base64 module discussed in lecture (Will have starter code)
b. Use the module base64 to read to the file

Provide Access to Code Exercises on Github