<https://swcarpentry.github.io/shell-novice/reference/>

pwd print working directory

ls list content of folder

ls –F list content of folder with type

man ls manual

q: quit

cd + name change directory to place

cd .. goes to directory with the file ..

ls –F –a shows also hidden files

cd back to home directory

absolute paths example cd/home/elise/Desktop/data-shell-data

cd ~ folder of user

cd - back to directory where you came from

mkdir make directory

rm remove file

nano filename.txt make file. Control X 🡪 exit

rm –r –i remove directory with interactive questions. Type Y if yes.

mv move file. Also able to overwrite Example: mv thesis/draft.txt thesis/quotes.txt

mv –I asks if you want to overwrite files if there already exists a file with the same name.

mv . move to directory where you are.

cp copy files

wc word count in file

wc –l count lines

> .filename.txt makes file from output

cat

sort –n sorts stuff inside on size numerically

| output of first command is used for second command

head –n 1 first file

tail –n 1 last file

“command” --help shows options for command. Example: sort --help

for filename in basilisk.dat unicorn.dat

> do

> head -n 3 $filename

> done

echo print

space in filename 🡪 put in between “ “, but “$filename” in a loop also

for filename in \*.dat

> do

> cp $filename original-$filename

> done

results in extra file with new name original-filename.dat

\*[AB].txt files end either on A.txt or B.txt

ctrl C interrupts process

history shows last commands

history | tail –n 5 shows last 5 commands

in a script use “$1” to do it for the first argument that is given

Run python with files in a specific folder: first change directory to the folder. Than start python

import numpy as np

np.loadtxt(filename, delimiter=’,’)

list: []

tuple: ()

Where to find python packages:

    - <https://pypi.python.org/pypi>

    - <https://docs.python.org/3/>

    - <https://bioconda.github.io/>

* google...

Matplotlib: <http://matplotlib.org/gallery.html>

import numpy as np

In [2]: import glob

In [3]: import matplotlib.pyplot as plt

In [4]: files = glob.glob('data/inflammation-\*.csv')

In [26]: def doPlot(file):

...: data = np.loadtxt(file, delimiter=',')

...: plt.imshow(data)

...: plt.title(file)

...: img = file.replace('.csv', '')

...: plt.savefig(img+'.png')

...:

OR

In [27]: for file in files:

...: doPlot(file)

In [28]: def doPlot(file):

...: data = np.loadtxt(file, delimiter=',')

...: plt.imshow(data)

...: plt.title(file)

SavedFile = 'images/' + file.replace('.csv','.png')

...: plt.savefig(SavedFile)

...: plt.close()

...:

In [29]: for file in files:

...: doPlot(file)

or

import os

os.path.splitext(file)

name, ext = os.path.splitext(file)

    plt.savefig(name+'.png')

os.mkdir(‘images’)

git config --global user.name "Elise van Bree"

git config --global user.email "elise.v.bree@gmail.com"

git config --global color.ui "auto"

git config –-list

go to directory

git init

>> Initialized empty Git repository in /Users/Elise/Desktop/planets/.git/

nano mars.txt

git add mars.txt

git diff shows difference between first and second version.

git add mars.txt

git diff --staged

git commit –m “message”

git log : shows all commits

git log -1 : shows only last commit made

git log --oneline shows on one line

git diff HEAD mars.txt compares to last commit

git diff HEAD~1 mars.txt compares to the one before the last one

git checkout HEAD mars.txt changes file back to the last file added to repository

git remote add origin [https link] put your repository on github.com

git remote –v check if it works

git push origin master put your repository online. Enter username and PW.

git remote set-url origin [url] change url.

git pull origin master get from online to your pc

git clone [link]