always use \, never /. / is considered as escape (still waits for command)

~ -> home folder

ls -> list the folders

cd (name od the folder) -> change direcotry (megváltoztatja azt a mappát, ahol dolgozok

cd .. -> goes up one folder

pwd -> shows that which folder you are in

cd . -> change to directory you are in(nothing happens)

cd -> without anything it takes you back the home folder

ls -F -> jobban kiemeli a foldereket egy /-val a nevek végén

ls -a -> feltünteti a rejtett fileokat is egy .-tal kezdődve

ls -l... -> bizonyos sorrendekben listázza a dolgokat (pl: -lt -> idő alapján)

man ls -> windowson nem múködik, de a szerepe annyi, hogy miket adjon ki millió és millió lehetőségekkel. csak man ls listázza a különböző lehetőségeket, ahogy szerkezteni lehet. más szóval az ls különböző lehetőségei. escape: press q

ls --help -> man ls windowson

cd doc (vagy bármi rövidítés, mappa kezdete) + tab -> ki fogja írni, ha bizonyos

mkdir (name) -> create a folder in the directory u are in (do not use áé and things like this, \_ instead of space, things like that)

rmdir (name) .> delete the named folder

nano (name) -> create txt file or any text file

cat (name of the file) -> it will present the content of the file

less (-.-) -> present in a different window t see more )exit: press q)

rm -r (name) -> delete directory, cannot be restored!!!!!!!!

***NEVER DO: RM -R /*** -> THE OS ALSO DELETED

rm -rf -> delete direcoty that cannot be deleted by default

touch (name) -> it will only touch the file )the modification date will change)

cp (name of the folder)/(name of the file) space (name of the folder where i want to copy)

cp – r (name of the folder) space (name of the folder) .> to copy for directories

cmd //c tree -> csak jól mutatja meg a fileokat a mappákban

vi -> opens some bullshit -> exit: :q! + enter

echo + sg -> like print in python

ctrl+C -> it will cancell any input or infinite loop even in python

grep -> ha elfeéejtek egy parancsot, de tudom, hogy minél használtam, akkor a historyban meg lehet nézni

.sh -> sima txt file, csak más neve van, viszont parancsokat lehet bele rakni (pl. megcsinálni 2000 vmit)

sh helyett lehet source is

sh -> shell script –parancsokat lehet bele rakni (pl. megcsinálni 2000 vmit)

adding variable: pl: f=5 +enter aztán echo $f -> ki fogja írni

printenv -> show all variable

git init -> létrehoz egy rejtett mappát

git commit -m "start tracking my shell script"

git status

git config --global user.name "Diamant Eszter"

git config --global user.email „[diamanteszter97@gmail.com](mailto:diamanteszter97@gmail.com)”

git status

git add (name of file)

git commit -m „explanation”

git log -> shows all of history happend in git folder

git check out (the long number given by command) ->

git pull -> any changes made on github will appear in the system as well

cloning -> **check in details**

git pull/push -> lehozni a módosításoat majd visszatölteni

git checkout -> change to master branch

git checkout experiment -> change to experiment folder