

Git for data science

feel to take notes here, they will be stored and uploaded to the event page after the course.  
<https://indico.mpi-cbg.de/event/149/>

Remember to say thanks to Peter for his work on the course :-)

MS Word - track changes

Google drive docs - previous versions available right side of window

First change your git settings

**git config --global user.name "Peter Pan"**

**git config --global user.email "Peter.Pan@tu-dresden.de"**

configuring the editor to use, use this link for a table of configurations versus multiple editors

<http://swcarpentry.github.io/git-novice/02-setup/index.html>

To see your git configuration : **git config --list** is the way to go

to see documentation:

**git help**

or

**git commandName -h**

.git's hidden folder created when you implement "**git init**"

if you delete this folder, there is no more version control

But if you still want to delete it

**rm -rf .git** (-r = recursive (includes removing subfolders), -f = force(doesn't ask if we really want to delete it))

**git status** to see current git

to remove unwanted gits in a folder

**rm -rf .git** (-r = recursive (includes removing subfolders), -f = force(doesn't ask if we really want to delete it))

open nano:

**nano mars.txt**

Show nano text: **cat mars.txt**

In nano, "write-out" = save

Write something to the git:

**git add mars.txt**

Setting u a commit message

**git commit -m ' i like trains'**

**git commit -m " yes u are"**

Seeing differences between last saved version and current unstaged version

**git diff**

Seeing differences between last saved version and staged version  
**git diff --staged**

What to add in the commit message - answer the question 'why' the edit rather than 'what' is the edit

good habit to keep adding whatever you want to do before the next commit in the staging area.

To add a commit without adding changes to staging area (NOT A GOOD HABIT)

**git commit -a -m 'commit message'**

**git commit -m 'commit message' filename.txt**

But care: You commit everything also from subfolders you mb didnt want to change!

For Windows Machines:

warning: LF will be replaced by CRLF in mars.txt.

The file will have its original line endings in your working directory

got to <http://swcarpentry.github.io/git-novice/02-setup/index.html>

to read the configuration how yo"u can tell git to automatically handle line endings

To look at log history

**git log**

**git log --oneline** (condensed manner)

**git log --oneline -1** (condensed and only the last 1 commits)

To unstage files ( remove from stage area):

**git reset HEAD <FILE>**

To compare current file to the previous saved version

**git diff HEAD mars.txt**

**git diff HEAD~1 mars.txt** (comparing with the version before teh last saved version)

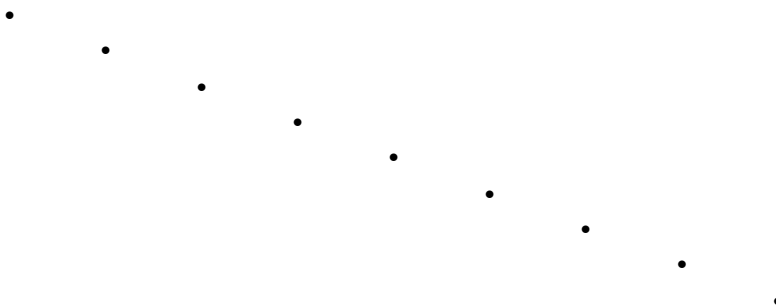
**git diff HEAD~4 mars.txt** (comparing with the version 4 times before the last saved version)

**git diff f96b86b..1a85080 mars.txt** (comparing hash1 to hash2 only for file mars.txt)

To restore your file to the last saved version

**git checkout HEAD mars.txt**

**git checkout HEAD ~13 mars.txt** (restores file to 13th previous saved version) You can also use the hash of a version explicitly



On  
ce  
you  
go  
bac  
k in  
tim  
e,  
you  
can  
als  
o  
go  
for  
war  
d  
unti  
l  
HE  
AD

**nano .gitignore** (enter names of files you want to ignore. You can use regular expressions as well)

**git status --ignored** (to see which files are being ignored)  
(remember to stage and commit the .gitignore file as well)

if you want to add a file that is being ignored, you can use the -f (force) handle

**git remote add origin2 https://link.com**  
**git push origin master**  
**git pull origin master**

h-pervez

To collaborate with someone, first clone their repository into your machine

**git clone https://collborators\_awesome\_repository.git newnameforrepository**

for those interested in resolving conflicts, see <http://swcarpentry.github.io/git-novice/09-conflict/index.html>

THX PETER FOR THE COURSE