Git Notes

Settings

git config --global user.name “HannatB”

git config --global user.email [j.a.tenbrink@uva.nl](mailto:j.a.tenbrink@uva.nl)

git config --global core.autocrlf input ##For mac/windows things

git config --global core.editor "subl -n -w"##text editor

check settings:  
git config --global –-edit

Start a repository

git init #in de folder with all the files (never remove .git!)

no need to make nested repositories (will cause problems)

Tell git which files to track changes and to commit changes

Git add filename

Git commit -m “Informative message”

Meaning of HEAD

Head is the most recent commited file! (and HEAD~1 the one before). If you change a file, it is not HEAD (until you’ve commited)

Restore previous files

Git checkout HEAD namefile #(without namefile, it will restore ALL old versions!)

Git checkout <unique id of last commit> namefile

* Git checkout will go away from where-ever you’re now (could be commited/uncommitted) and go to where you point it to go (so HEAD will go to latest commit, HEAD~1 previous, etc).
* If changes are uncommitted, you will loose them (I think)

Unstage files

If you’ve added files but not yet commited, checkout will NOT work. First necessary to reset the staged files via

Git reset HEAD filename #unstages files!

Git reset filename #unstages files denk ik ook

Ignore files

Create (and add&commit) a .gitignore file. Here, you can add all files (or folders) to ignore, and also include exceptions by using !

Note that all previously commited files will NOT be ignored, even when they follow the rule!

Remote

Bitbucket, daar een nieuwe repository opzetten zonder files,   
Aan git laten weten waar die remote zit (git remote add origin git@bitbucket.org:HannatB/NAME.git)  
en dan push/pull

* git pull/push origin main

Conflicts:

Avoid them by first pulling from remote, after making changes push them. If still happens, first pull, then change the affected files by adapting the conflicts. And after that push