Introduction to Git & Github

Notes introduction

Git repository = timeline: Everything is stored here, however if you go back and adjust something, the 'present' is also adjusted. It is locally stored and you can work together with other people.

Github = Backup of your timeline: Instead of going back in 'time' to adjust your document, you just retrieve a copy of your old document that you can adjust, without adjusting the newest version of it. (also online repository for code). Backup place for Git project. Cloud based location ~remote

- git init to initialize git -> only sees whats in folder and under!
- git is invisible in folder. Its the history of all the things you do in the folder
- Any point in timeline is a snapshot: When/how to take snapshot? Develop area -> staging area -> local repository area (commitment!)
- $ git commit
- Everytime you commit, git makes you write a message of why the snapshot is made! So make it a meaningfull snapshot/message
- $ git commit -m "meaningfull message"
- What is meaningfull? Why was it changed/ How this addresses the issue / Effects due the change / Limitations of the change (eg i changed part of code to make it work but now other part doesnt work / I studied the main thing but now i need to study something else to understand the main part)
- Be as descriptive as possible! More info is better than too little
- Git is tracking so just tell what y-ou mean with the changes you made, no need to state where you changed or what you put in
- IMPORTANT: don't double ´´´git git´´´
- IMPORTANT: dont double init init git
- You can save as much as you want but it won't create a backup in timeline unless you commit!

- Don't init git in a subfolder where the parent folder already init git
- Init in subfolder will create a timeline/repository inside a timeline/repository

Conceptual areas

- Develop area: working directory
- Staging area: Temporary space. Place where you dump stuff before putting it in local repository (git add). Also to organize logically your area. You can add 1 file, commit, then add another and commit and they will both be saved in the timeline as 2 seperate timespot.
- Local repository: Place where your snapshots/timeline is saved
- Remote repository (in our case it is github)
    - Create repository
    - Name: Easy if you use the same name as your repository (the name of folder that git is tracking)
    - Description
    - README.txt: Detailed description of your project and tool usage (if it is the case) (should always have one!) Github will automatically recognize the file README.txt and show it on frontpage.
    - .gitignore: In order to ignore files/lists

      Make a .gitignore and in it list all files that should be ignored
    -

Git status allows me to check what files are:

- To be staged: You have committed, it before, you have made new changes and git recognise the new changes are not yet added nor committed
- To be committeed: You have committed this file before, you have made new changes and git recognises you have added but not yet committed
- untracked files: Is a completely new file/folder, you have made changes and git recognises that you have not yet added nor committed.
    - This way you can check and organise in combination with staging area. Everything in staging area will be comitted together.

Travelling in the timeline (local repository)

- git log to see history of all your commits: who, when, msg, commit ID
- git show/diff to see difference between two commits
    - Show just shows what happend in ID1 and then ID2
    - Diff shows the difference between ID 1 and ID2
-