

Version control management and research collaboration using git and github

An introduction

APSYS group

MCC Berlin

July 11th, 2019

What is git and GitHub?

Why should I use it?

How can I use it?

What is version control management?

Software to keep track of the history and different versions of files within project folders

What is git?

- git is a program for version control
- designed for distributed software development
- created by Linus Torvalds for his work on the Linux kernel

Explain idea of a git repository

What is GitHub?

Explain the idea of a remote repository

Explain github (and providers of remote repositories like gitlab, bitbucket, SourceForge, Launchpad ...)

What is git and GitHub?

Why should I use it?

How can I use it?

Why use version control in research?

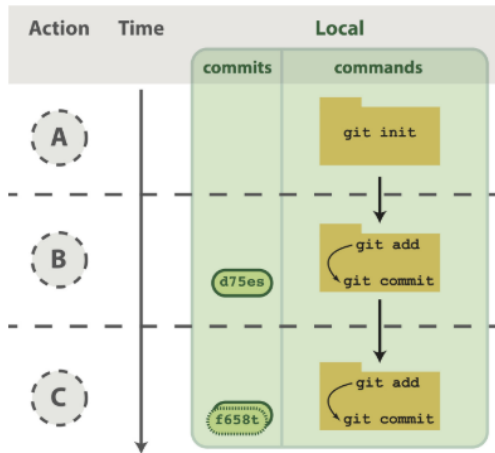
- getting some order in the mess
 - data
 - software code/scripts
 - manuscripts for papers
- sharing your code or
- collaboration and attribution of work

What is git and GitHub?

Why should I use it?

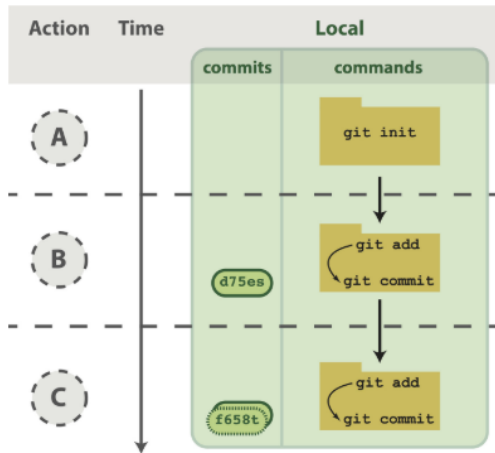
How can I use it?

Git Workflow (simplest)



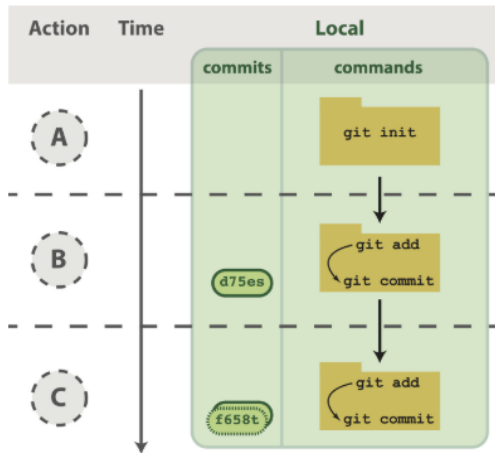
- Keep track of changes in a folder on your computer

Git Workflow (simplest)



- Keep track of changes in a folder on your computer
- Changes are stored as lines added and removed

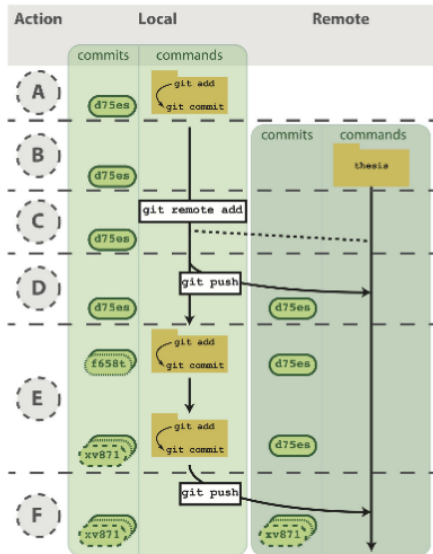
Git Workflow (simplest)



- Keep track of changes in a folder on your computer
- Changes are stored as lines added and removed
- No need to save multiple versions of the same file; you have recorded all changes and can view or revert these at any time

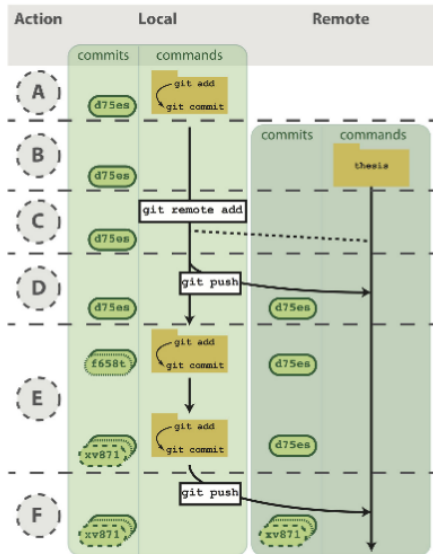
Git + Github Workflow (simplest)

- Attach your repository to a remote version



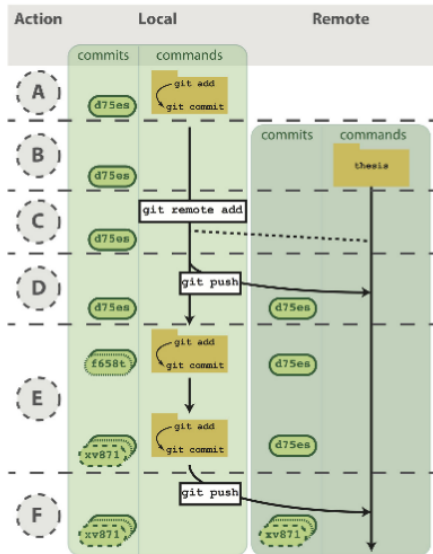
Git + Github Workflow (simplest)

- Attach your repository to a remote version
- If working with collaborators, they also can make a copy (clone) on their machine

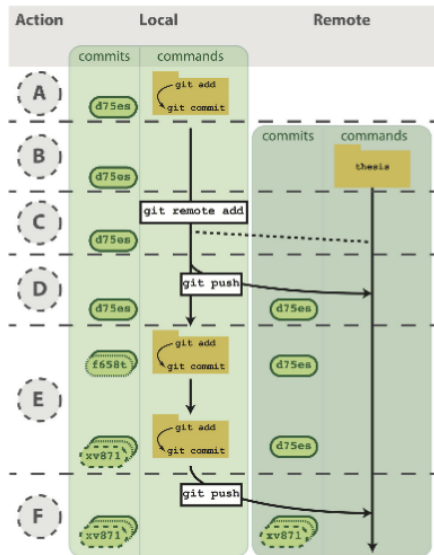


Git + Github Workflow (simplest)

- Attach your repository to a remote version
- If working with collaborators, they also can make a copy (clone) on their machine
- By both using pull, you can keep up to date with each others' changes



Git + Github Workflow (simplest)

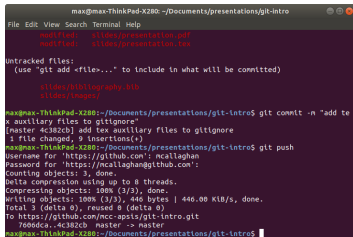


- Attach your repository to a remote version
- If working with collaborators, they also can make a copy (clone) on their machine
- By both using pull, you can keep up to date with each others' changes
- For more complicated workflows, especially where maintaining a working version is critical, check out branching <https://guides.github.com/introduction/flow/>

Tools

Command Line

- Easy to document/explain



```
max@max-ThinkPad-X280: ~/Documents/presentations/git-intro
File Edit View Search Terminal Help
modified: slides/presentation.pdf
modified: slides/presentation.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)

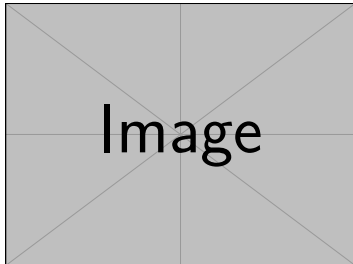
  slides/bibliography.bib
  slides/images/

max@max-ThinkPad-X280:~/Documents/presentations/git-intro$ git commit -n "add tex
x auxiliary files to gitignore"
[master 4c382cb] add tex auxiliary files to gitignore
2 file changes, 9 insertions(+)
max@max-ThinkPad-X280:~/Documents/presentations/git-intro$ git push
Username for 'https://github.com': mcallaghan
Password for 'https://mcallaghan@github.com':
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 446 bytes | 446.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/mcc-apsis/git-intro.git
  7606dca..4c382cb master -> master
max@max-ThinkPad-X280:~/Documents/presentations/git-intro$
```

Steeper learning curve, but more flexible and harder to do things unintentionally

GUIs

- Easy to use



Often there are integrations in development environments, e.g. RStudio, Atom

Starting a repository

To start working with a repository, either turn an existing folder into a git repository

```
git init
```

or copy an existing repository into a folder

```
git clone
```

Managing the repository

git status

git diff

staging:

git add [-all]

git commit [-a]

Branches and merges

explain what branches are

`git branch`

`git merge branchname`

`git checkout`

Interacting with remote repositories

`git pull`

`git push`

Warning: careful with copyrighted materials in public repositories
forking and pull request for working on repository for which you are
no collaborator

Further useful commands and tools

.gitignore file

create doi for citations:

<https://guides.github.com/activities/citable-code/>

Questions?

Practice / task

- clone remote repository with
`git clone https://github.com/mcc-apsis/git-intro.git`
- add some question or feedback to the presentation in the file
- add and commit changes
- pull changes already made by other
- push your own changes

References