



Version Control, MarkDown and Social Coding

Cyril Pernet
Robert Oostenveld

Intended Learning Outcomes

- Describe what version control is
- Demonstrate basic Git and GitHub command knowledge
- Contribute to 'code' in social context (a.k.a. social coding)

Vresions

version

Version

Versions

How many of you have written a paper that has been peer reviewed and published?

How many versions of the manuscript did you need for that?

What are the similarities and differences in writing code and writing text?

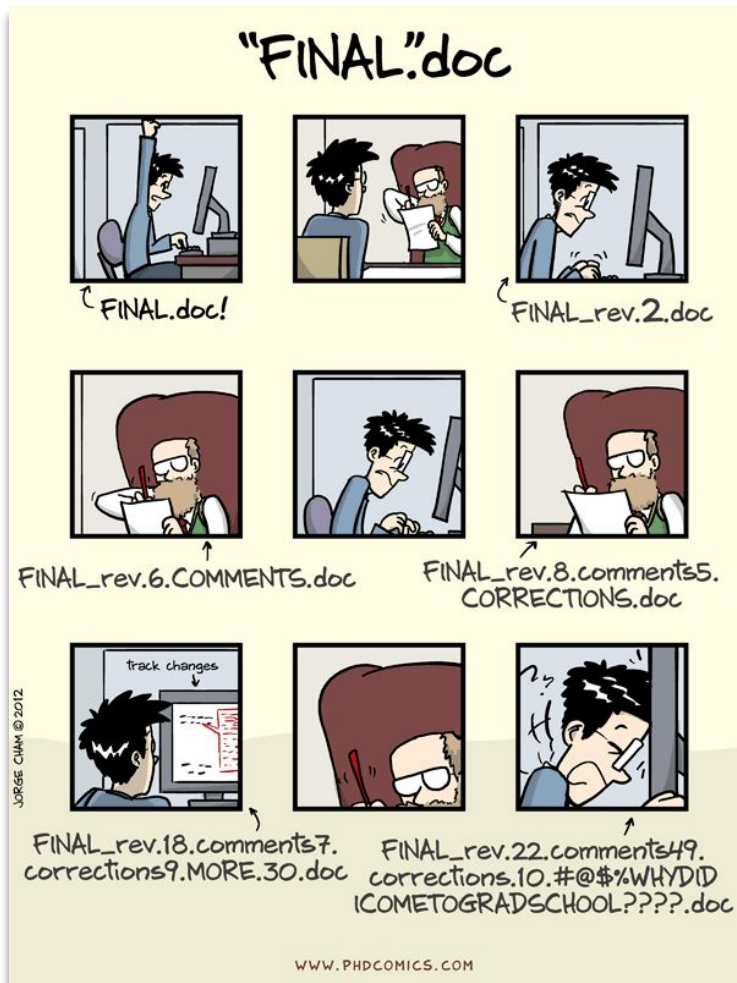
Teacher notes: use some platform for students to share (preferably allow anonymous input for those not feeling comfortable with that - also nothing is mandatory here)

Versions

How many of you have written a paper that has been peer reviewed and published?

How many versions of the manuscript did you need for that?

What are the similarities and differences in writing code and writing text?





git

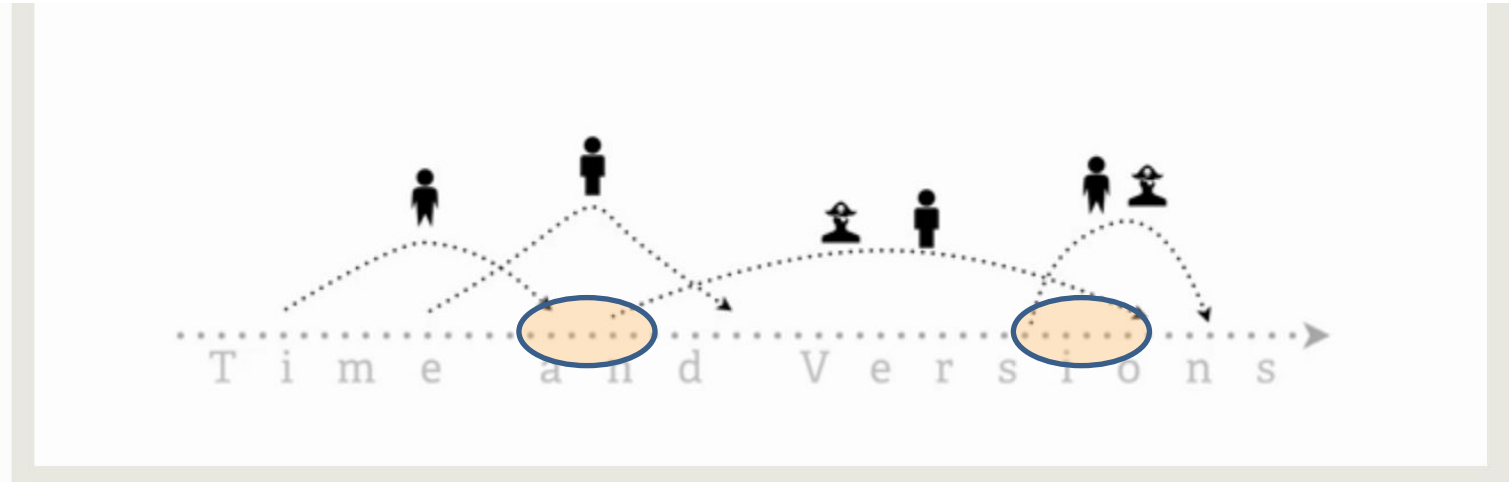


GitHub

<https://git-scm.com/video/what-is-version-control>

The timeline

- As explained in one of the Git videos, the time line of file editing looks something like this



Overlapping time – two people working on the same file -
merging and conflict resolution (GitHub magic)

Practical PreRequisite

- 1 - Get a decent editor Visual Studio Code is recommended
- 2 - Go to GitHub and open an account (if you haven't)
- 3 - Windows and Mac users, download and install GitHub desktop (GUI to manage repositories, it really is easier) → this will install Git as well

Linux user? Nothing to do :-) git should be already installed

---> we are going to use command lines to learn how it works -- GitHub desktop and VSCode can do this for you

Step 1: let's link stuff

```
git config --global user.name "GitHubUserName"
```

```
git config --global user.email "YourEmail"
```

Step 2: let's make a repository

```
cd somewhere
```

```
mkdir somefoldername
```

```
git init
```

```
ls -la
```

```
cd .git
```

```
ls -la
```

```
cd ..
```

Step 3: let's add your markdown from earlier

cut/paste that file into your new folder

`git status`

`git add .`

`git commit -m "my first commit"`

Change something in your markdown

`git log --oneline`

`git restore yourfile.md`

Step 4: let's not track

Create a file

```
touch tmpinfo.txt
```

Create a folder and file notes/[notes.md](#)

```
mkdir notes
```

```
cd notes
```

```
touch notes.md
```

```
cd ..
```

Now create a .gitignore file, and let's edit it as to ignore what we want to ignore (use wild card for folder contend)

Time to get online!

Head to GitHub and **fork** the course repository

In GitHub desktop, pull the repository, and in /version_control copy/paste your markdown file, then time to push

→ git push

Now do a Pull-Request !!! what is this magic

(let see online, how we manage pull request, vs locally, and how can we merge all your files into ours - at the end we should have our markdown as before but all names in the list)

Branches, forks, issues

Let's try stuff together and edit this [MyFirstMarkdownFile.md](#)

- a brave soul coming to show what happen if one clone instead of fork?
- a brave soul to try a branch
- a brave soul to raise an issue and branch