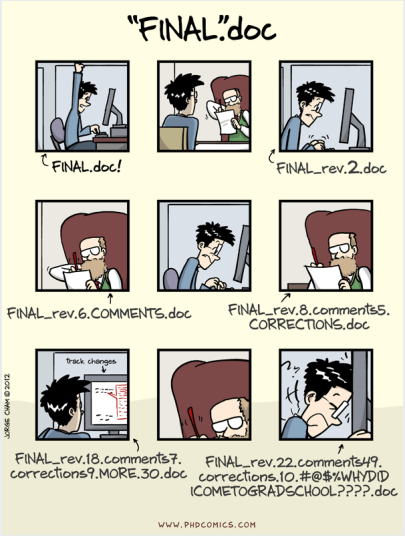
Version control with Git (and GitHub)

September 28, 2021

MathSys - Intro to Computing 2021/22

Why version control?

- To keep track of changes on several files (e.g. code, text, presentations).
- To go make to older versions, undo mistakes.
- To work collaborately (branches) and compare changes.
- To keep a secure copy as a back-up (GitHub).



Getting started

We will follow a simple guide.

1 - Download Git

Then check:

```
$ git --version
```

2 - First setup

Configure Git with username and email address

```
$ git config --global user.name "your-username"
```

```
$ git config --global user.email "your-email-address"
```

3 - Configure a GitHub account

Go to the GitHub website.

Exercise 1 - Tracking changes

Create a repo

- Let's create a directory in `Desktop` and move in that folder.
- Initialize a repo:

```
$ git init
```

- Check there is an inner Git directory:

```
$ ls -a
```

Make and add changes

- Use `vim` or `nano` to create a txt document and save it as "file1.txt".
- Check the status of the repo

```
$ git status
```

- Add the file to the repo

```
$ git add file1.txt
```

```
$ git status
```

- Finalize your updates with a commit

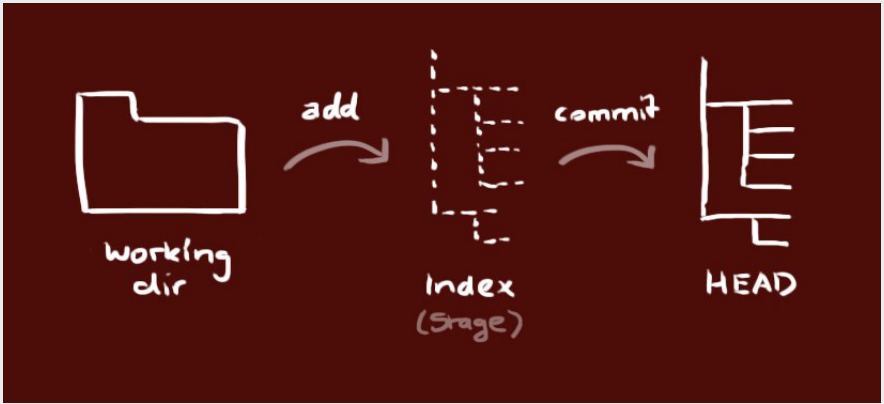
```
$ git commit -m "your-message"
```

Further commands

```
$ git log
```

```
$ git diff
```

The workflow



Exercise 2 - Remotes in GitHub

01

clone*

Create a copy of repo from remore server

```
$ git clone https:/github/path
```

or *fork* a repo on GitHub.

02

push from remote

Add existing repo to GitHub

```
$ git remote add origin <REMOTE_URL>
```

```
$ git push origin main
```

further reads here

03

add changes and push

Make and add changes as before, then copy them from local to remote:

```
$ git push origin master
```

04

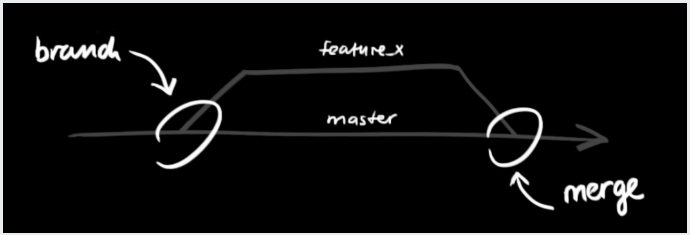
pull

You can also copy changes from remote to local: `$ git pull`

*Try with https://github.com/mathsyscomputing/IntroToComputing.

Branching

You might want to develop chunks of code separately or even change some main features.



`$ git branch <new-branch>` <- create a new branch

`$ git checkout <new-branch>` <- switch to it

If you are sure, `$ git merge <branch-to-add>` merge branch with your current one.

Some useful resources

- Online book on Git
- Version control with Git lessons
- GitHub Docs

Some cheat sheets:

- Git cheat sheet
- GitHub/Git cheat sheet (direct download)