

# Introduction to Git using GitHub

Modern Techniques in Modelling

LONDON  
SCHOOL of  
HYGIENE  
& TROPICAL  
MEDICINE



# At this point you should have...

## 1. Checked if git is installed

- Go to a Command Prompt window and type:
- `git --version`
- If not, then go to: <https://git-scm.com/download/> and run the download.
- Accept all default options.

## 2. Created a GitHub account: [github.com/join](https://github.com/join)

## 3. Downloaded GitHub Desktop: <https://desktop.github.com/>

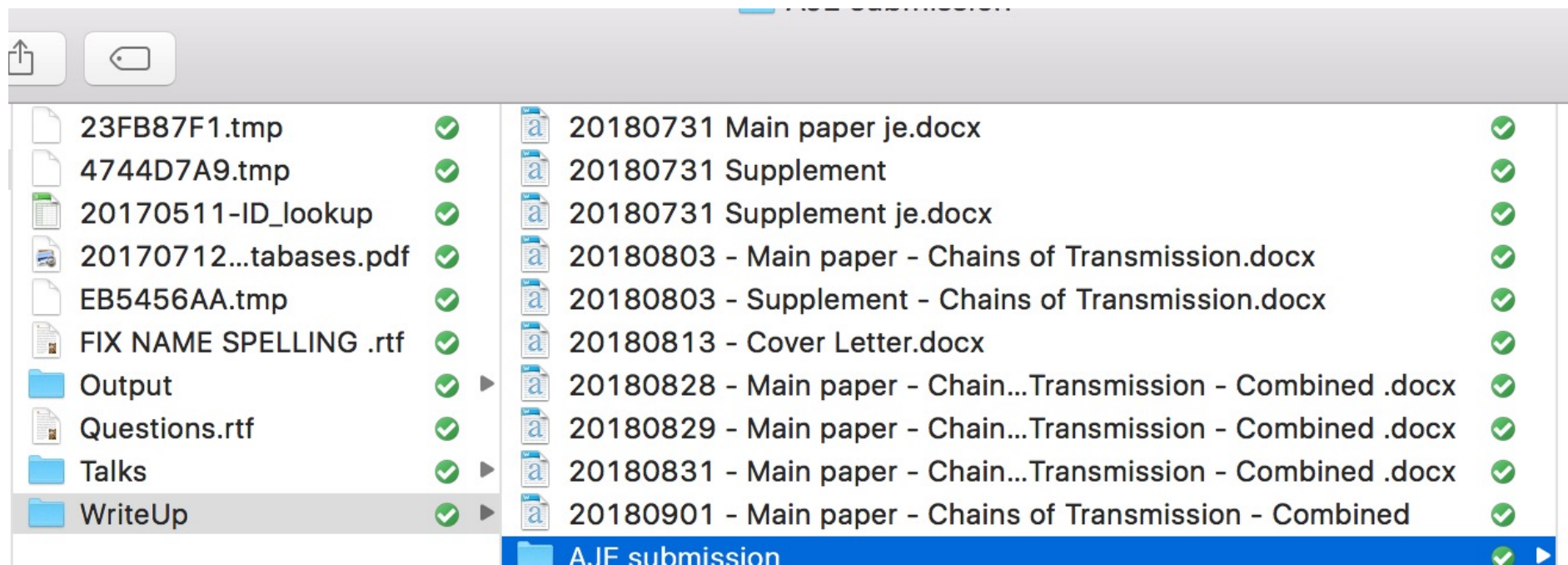
- Open it, and sign in using your GitHub account

# Objectives for this hour

- Introduce the idea of version control – for code
  - What’s Git? What’s GitHub? What did I just install on my computer?
- Familiarise yourselves with some “infrastructure” of Git and GitHub
- Introduce some interfaces that can make it easier to use Git and GitHub
- Learning by doing!

# What is version control?

- A system that records changes in files
- Everybody has used it, even if it's a system you have made yourself!

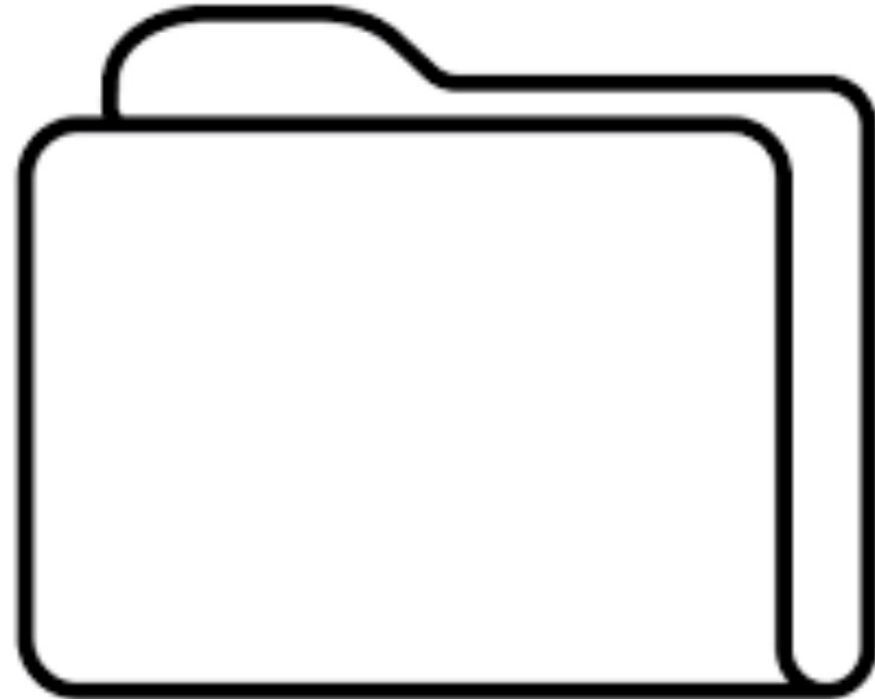


# Why use version control?

- Backup/restore/saving
- Synchronisation
- Track changes (and ownership)
- Seeing your progress
- Professional way to create code

# The most common version control system is Git

- What is git?
- What is a repository?
- What is a commit?
- What is github?



# Why?

## Git

- Flexible
- Remembers EVERYTHING
- Conserving and preserving
- Easy comparison

## GitHub

- Remember EVERYTHING (online)
- Public and private repositories (<- open science!)
- Easy collaboration
  - The “issue board”
  - Ownership
- Showcase your work!

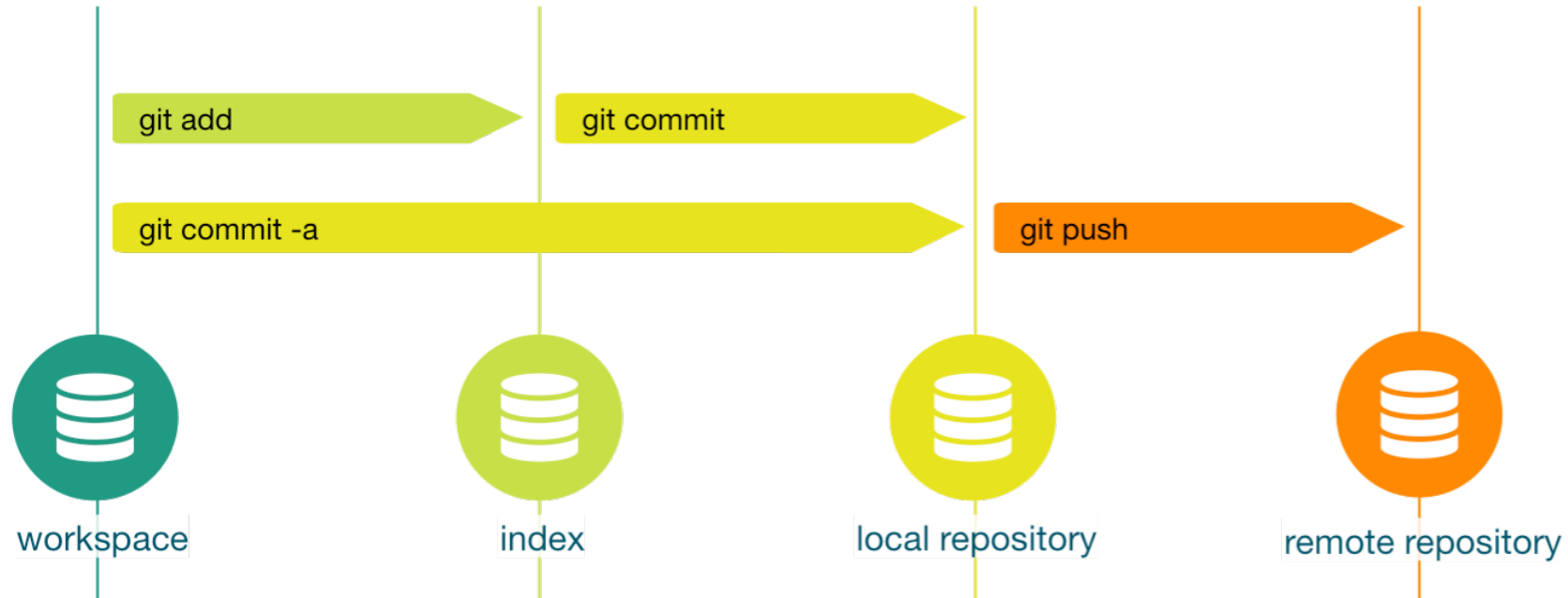


There are a lot of new words! It's just lingo, do not be afraid!

- Add
- Commit
- Push



# git paradigm

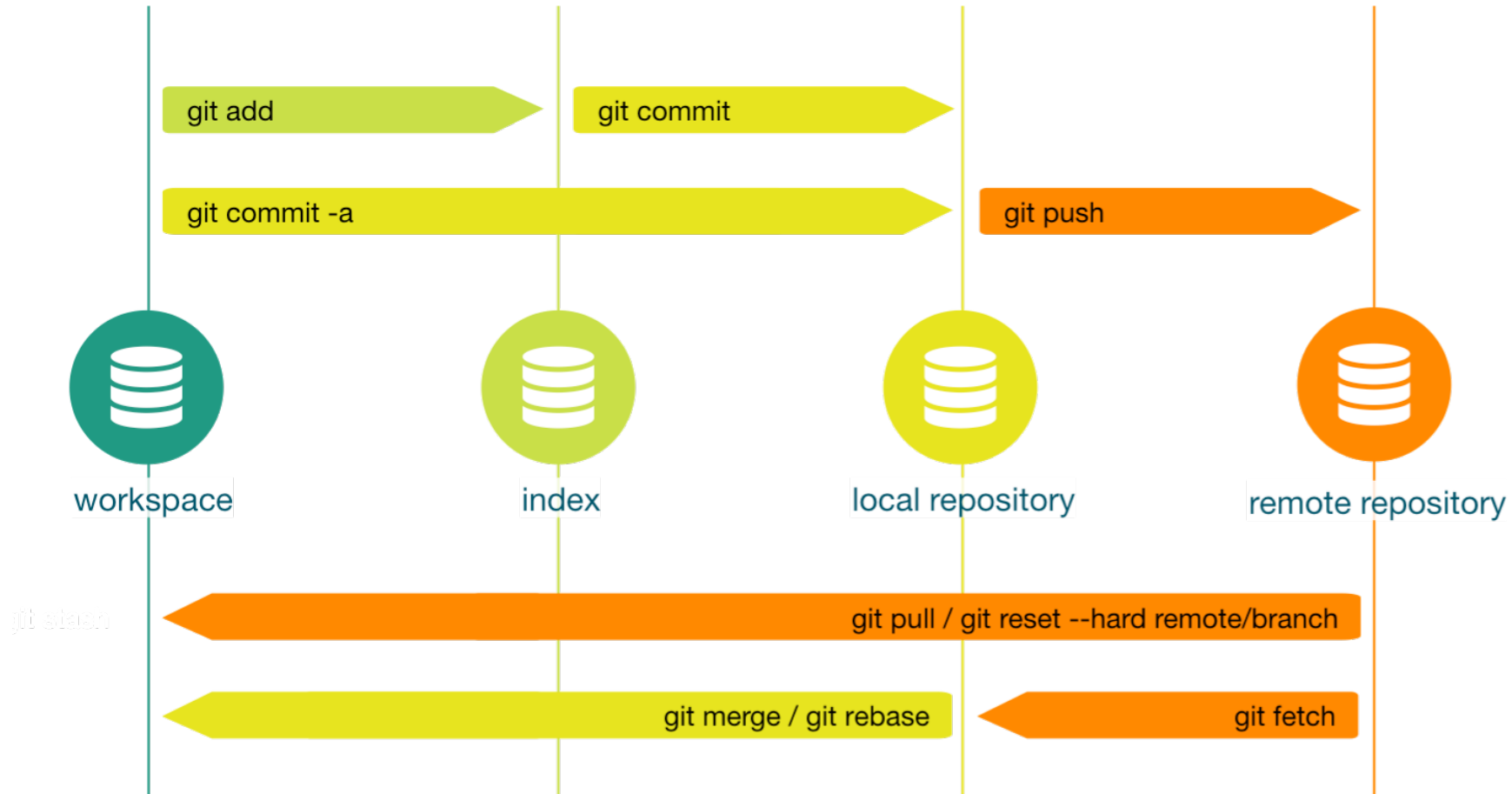


`git st`

There are a lot of new words! It's just lingo, do not be afraid!

- init
- clone
  
- Add
- Commit
- Push
  
- Fetch
- Pull
- Merge

# git paradigm



# Using an app to interact with Github

- Recommend GitHub Desktop (free, easy-to-use)
  - There are many others (e.g. inside RStudio)
  - Benefits: getting started, easy-to-understand, overview
- 
- What does it do?
    - "watches" the repository
    - Tells you about changes to the files in the folder
    - Helps you commit the changes

# Some key dos and donts

## Do:

- Commit often
- Commit small changes as you go
- Write useful commit messages

## Don't:

- Commit untested changes
- Add data files
- Add untrackable files
- Add enormous files

# Best practices

- .gitignore
- Add a README
- Add a License
- Do not put a repo in Dropbox

# Further resources

- Lots of online tutorials, e.g. <https://programminghistorian.org/en/lessons/retired/getting-started-with-github-desktop>
- Some videos e.g. <https://www.youtube.com/watch?v=77W2JSL7-r8>
- Learn by doing!