# Beginner tips from an unskilled user for using Git and Github for Version Control

Lars Nesheim

November 2023

# What is Git?

- **Git** is a program for simple, efficient **version control**:
  - Records changes to files, sets of files.
  - Enables efficient management and control of multiple versions of files.
- Can be used to:
  - Compare **multiple versions, track changes** over time, track changes by specific users.
  - Store archive or **"repository"** locally, centrally, or **in cloud** (e.g. Github).
  - Archive specific versions and/or revert to previous versions.
  - Synchronise versions across **multiple computers and multiple users.**
  - Create **branches** of projects.
- Works on Linux, Mac, Windows.

# What is Github?

- Github is a **cloud platform** owned by Microsoft.
- Allows for cloud storage of Git repositories.
  - Can be accessed from any computer with web access.
  - Store project repository in cloud.
  - Can easily synchronise with local computers.
- Allows for either private or public repositories.
- Many tools for code development, collaboration, project management.
- Alternatives include Bitbucket and many others.

# Getting started

- Using Git and Github for the first time.
  1. Install Git (if necessary).
     - By default, Git is installed on Linux and Mac (not sure about Windows).
     - On Windows, can download/install Github Desktop.
  2. Setup Github account.
     - Free personal account has most basic tools/services.
     - UCL account has additional benefits/tools/services.
  3. Set up secure access to Github.
     - Use ssh keys.
     - These function like a password to enable password protected access.

# Starting a project

1. Create repository on Github
2. Clone repository from Github to local computer.
3. Add, delete, or edit files on local computer.
4. Commit changes (with a brief message describing changes).
5. Push changes from local computer to repository.

# Work on existing project

1. Assume local computer already contains copy of remote repository.
2. If necessary, commit any changes to files on local computer.
3. Pull files from remote repository.
   - Option to switch to a different branch.
4. If any conflicts, resolve them, commit changes (with message), and push to remote repository.
5. Add, delete, or edit files on local computer.
   - Option to create new branch.
6. Commit further changes (with a brief message).
7. Push changes from local computer to repository.

# Getting started

- If desired, download **Github Desktop**.
- Setup a **Github account**.
- Setup secure access.
  1. Generate **ssh key** and add to your computer's **ssh-agent**. **Instructions**.
  2. Add the ssh key to your Github account. **Instructions**.

# Starting a project

- Create repository on Github (name, pubilc/private, readme, template, license). Edit readme.
- Clone repository from Github to local computer.

```
git clone git@github.com:larsnesheim/Github.git
```

  - **command:** `git clone`
  - **address:** `git@github.com:larsnesheim/Github.git`
  - All files in repository are copied to local computer in new directory "Github".

# Starting a project: continued

Table 1: Other commands

| | |
|---|---|
| `git status` | Check on current status. |
| `git add newfile.txt` | Add new file to project. |
| `git rm oldfile.txt` | Remove old file. |
| `git commit -a -m "Describe changes made."` | Commit changes. |
| `git push` | Copy changes to remote re |

- **Final two steps (1) commit, 2) push) are crucial.**

# Work on existing project

- `git pull` to copy most recent version of repository from remote to local computer.
- `git checkout -b new_branch` to create a new branch and switch to it.
- `git checkout main` to switch to main branch.
- `git push -u origin new_branch` to copy new branch to remote repository.
- `git checkout --track origin/branch` to switch to and track "branch".
- `git merge branch_to_merge` to merge current branch and "branch_to_merge".

# Things to always avoid

- Never copy large data files to remote repository.
- Never copy confidential data to remote repository.
- Never copy passwords, API tokens, or other confidential information to remote repository.

# Other topics

- Using Github desktop.
- Using Github.
- Using submodules.
- INntegrated with Matlab, RStudio, many IDE's, Overleaf.
- Finding help:
  - Search on internet.
  - **Github documentation**.
  - **Git documentation**.