Version control with `git` + `ssh` and collaborative software development with `conda`

Florian Schunck

Installation 🛠️

⚠️⚠️⚠️

For everything that follows **stackexchange** and **chat-gpt** are your friends

⚠️⚠️⚠️

For the workshop we need the following software

1. `git` for version control
2. `ssh` for secure (and automated) communication with remote servers
3. `conda` for reproducible environments

Install `git`

- Windows: https://git-scm.com/downloads (and just keep all the default settings). This will also install SSH
- Linux (debian): `sudo apt install git`
- Mac: Install homebrew (see https://brew.sh/) and then install git with `brew install git`.

Become familiar with the tool

- Open a commandline interface and type `git --help` to see if everything works. Then look at `git clone --help` to understand what happens when you clone a repository

Install `ssh`

`git` can be used via unsecured connections, but many applications require the use of encrypted, secure connections. SSH (*s*ecure *sh*ell) operates by exchanging a public and private key between yourself and the service that you want to connect to.

Install `ssh`

Generate an **SSH keypair** with openssh and upload to your github account (Installation guide: https://www.geeksforgeeks.org/how-to-add-ssh-key-to-your-github-account/)

- ⚠️ **First**, Test if `ssh` is already available by typing `ssh --help` in your CLI
- Linux: ✅ Usually already installed. If not: `sudo apt update && sudo apt upgrade` to update the package repositories, then `sudo apt install openssh-server openssh-client` to install ssh
- Mac: ✅ Already installed
- Windows: ✅ In modern windows versions SSH should be preinstalled. You can verify it by typing `ssh` into cmd. If not use ssh that comes with the git bash CLI.

Generate Keys

1. Creating a key-value pair with ssh: `ssh-keygen -t ed25519`
2. Log into the https://github.com
3. Copy the public key and paste it in the respective section of your github account

Reproducible software environments

Science can only be advanced if the results of previous works can be reliably reproduced. This is true for wet-lab experiments as well as for dry-lab experiments (i.e. models). **In order to make a workflow reproducible, we have to tell others what the requirements are that we run the software in.**

Conda is a package manager 📦 that facilitates this.

First, see if conda was already installed `conda --help`, if not: Follow the instructions on: https://docs.anaconda.com/miniconda/install/#installing-miniconda

Practice 🛠️

Go to: https://github.com/flo-schu/collaborative-software-development and follow the README

Everything in the README worked? Done

✅ You are ready to go 🚀