Please note!

- By attending this class, you consent to being recorded.

- This recording will be placed in an online folder accessible to the students of this class. It may also be distributed to other DTU students for education purposes.

46120: Scientific Programming for Wind Energy

Git and GitHub Teaser

Jenni Rinker

Agenda for today.

- Pull new course material ✔

- Meet the 46120 Teaching Team. ✔
- Course introduction: Jenni. ✔
- What is good code: Ju Feng. ✔
- Teaser for git/GitHub: Jenni.

- Begin groupwork on Week 1 homework.

Teaser for Git and GitHub

Version control is great.

Scientific programming requires version control.

Scientists and engineers develop code to load, generate, analyse, model, and/or visualize data.

This code is often extremely *dynamic*.

- Fixing bugs, implementing new features, etc.

Common but suboptimal way to track these changes:

```
project_code/
    analyse_data.py
    analyse_data_v2.py
    other_analysis.py
    other_analysis_v2.py
    make_plots_FINAL.py
```

Difficult to track history, revert changes. Further, does not allow for collaboration.

- Git and GitHub (or GitLab/BitBucket) are tools that address these issues.

Git is a distributed version control software.

- “Version control software”:
  - A program that tracks who made what changes to which parts of files.
- “Distributed”:
  - Copies of the files & history can be located on different computers.
  - Changes in different computers can be transferred and merged.
- Git is by far the dominant software in use. You will use it throughout the semester in this course.
- Unlike OneDrive/Dropbox, checkpoints are done manually.
- A collection of files whose history is tracked is called a “**repo**” (repository).

Overview of the git process.

working directory — *a bunch of files with changes, but only a few are ready to be added to history.*

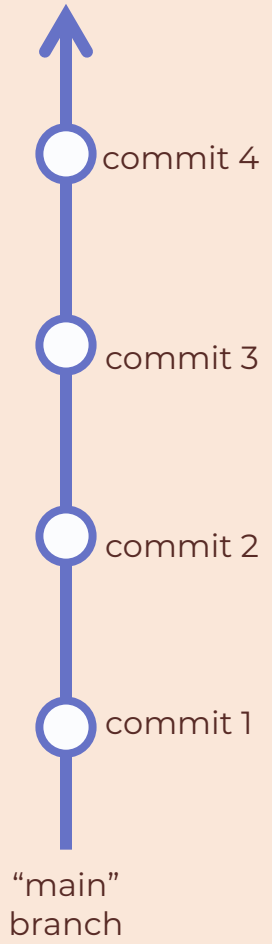
git add ← *tell git what files are ready*

staging area — *the files in the staging area are ready to be saved to repo history.*

take a snapshot! → **git commit**

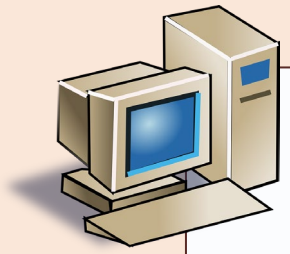
repository — *the repository has the history of the files.*

Image from here



GitHub is a hosting service, an online version of your repo.

Git versus GitHub.



Git

ORIGIN – your computer

Git terminal

GitHub

REMOTE – online server

https://github.com/

status	*check current status*
add	*add files to staging area*
commit	*save "snapshot" of repo state*
clone	*download repo from server*
pull	*update local copy from server*
push	*update server from local copy*
branch	*create a new branch*

Excellent git cheatsheet: here

Common git commands.

Run these in the Anaconda Prompt or git-scm terminal.

- Download a repository from GitLab (“cloning”):

```
cd <directory where you want the repository>
git clone <path to github/gitlab url>
```

- Update your local copy from the cloud (“pulling changes”):

```
git pull [origin main]
```

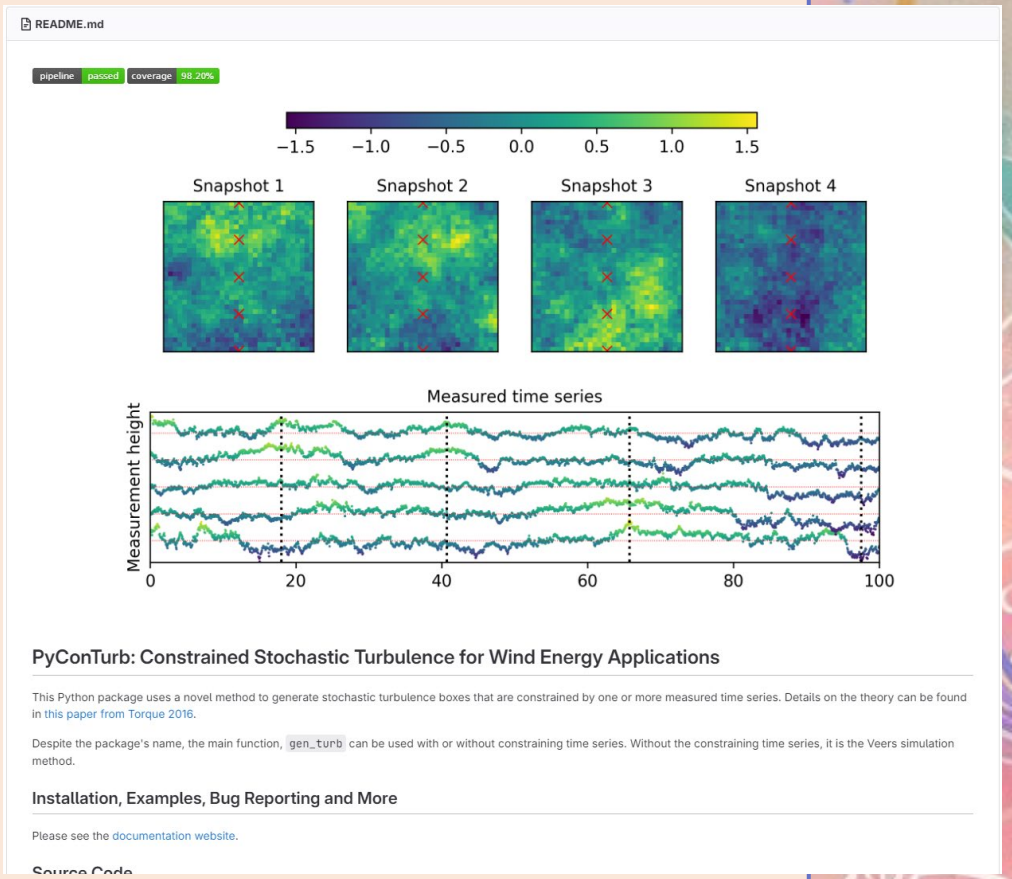
- Update the cloud from your local copy (“committing and pushing”):

```
git add <file1> <file2> <folder1>
git commit –m "<insert commit message>"
git push [origin main]
```

Push to cloud

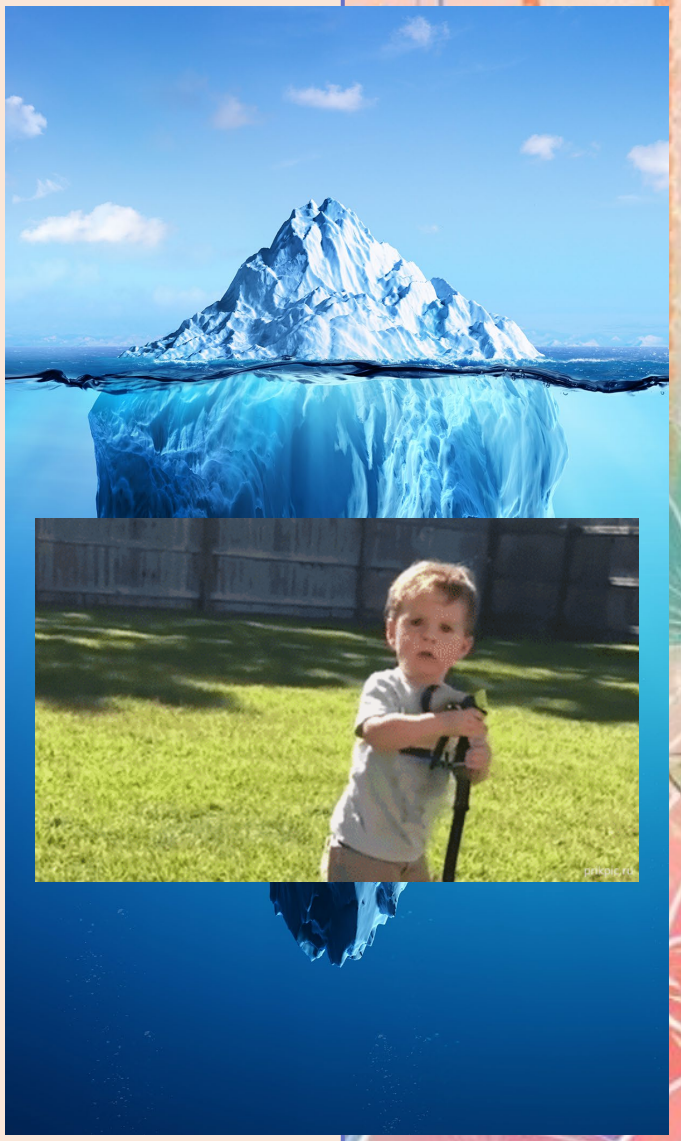
Example of GitLab/GitHub repositories in the wild.

- TOPFARM.
  - A Python package for wind-farm optimization.
- PyConTurb and Hipersim.
  - Python packages for creating turbulence fields from measurements.
- IEA Reference wind turbines on GitHub:
  - 15 MW.
  - 22 MW.
- Open-science example of a repo containing code for a paper.



Final remarks.

- Git has a lot of features/complexity.
  - Only basics here. And multiple ways to do same thing.
  - Be aware when searching the internet.
- Be very, very patient with yourself.
  - A lot of new things: command line, git, GitHub, etc.
  - This week may feel overwhelming. It slows down, promise.
- Seek support when needed.
  - Try teammates or Google to solve issues first, but of course utilize Slack, office hours and/or TAs.
- NEVER clone a repo into an existing local repo.
  - All repos should be in separate folders on your computer.



Homework for this week

Time to get your hands dirty!

Start the homework.

*Remember, you're expected to work **about 6 hours outside of class**. Schedule time for that.*

- A full list of homework tasks is in the Week 1 folder on the Course Material repo:
  - 46120-PiWE/week01_intro_git at main · DTUWindEducation/46120-PiWE

In a moment, we will open BORs, one for each team.

- Each team enters their BOR (same as your team ID). Perhaps find a physical spot outside the auditorium.
- Complete **Part 1** of the weekly assignment.
- If there are issues joining your team repo on GitHub, contact a TA/instructor today.

Any questions?