Introduction to version control with Git

Day 3: More Git

Selina Baldauf

November 25, 2024

Today

- Collect open questions in Excalidraw (5 min)
- Discuss open questions and additional topics (40 min)
- Feedback and Goodbye (10 min)

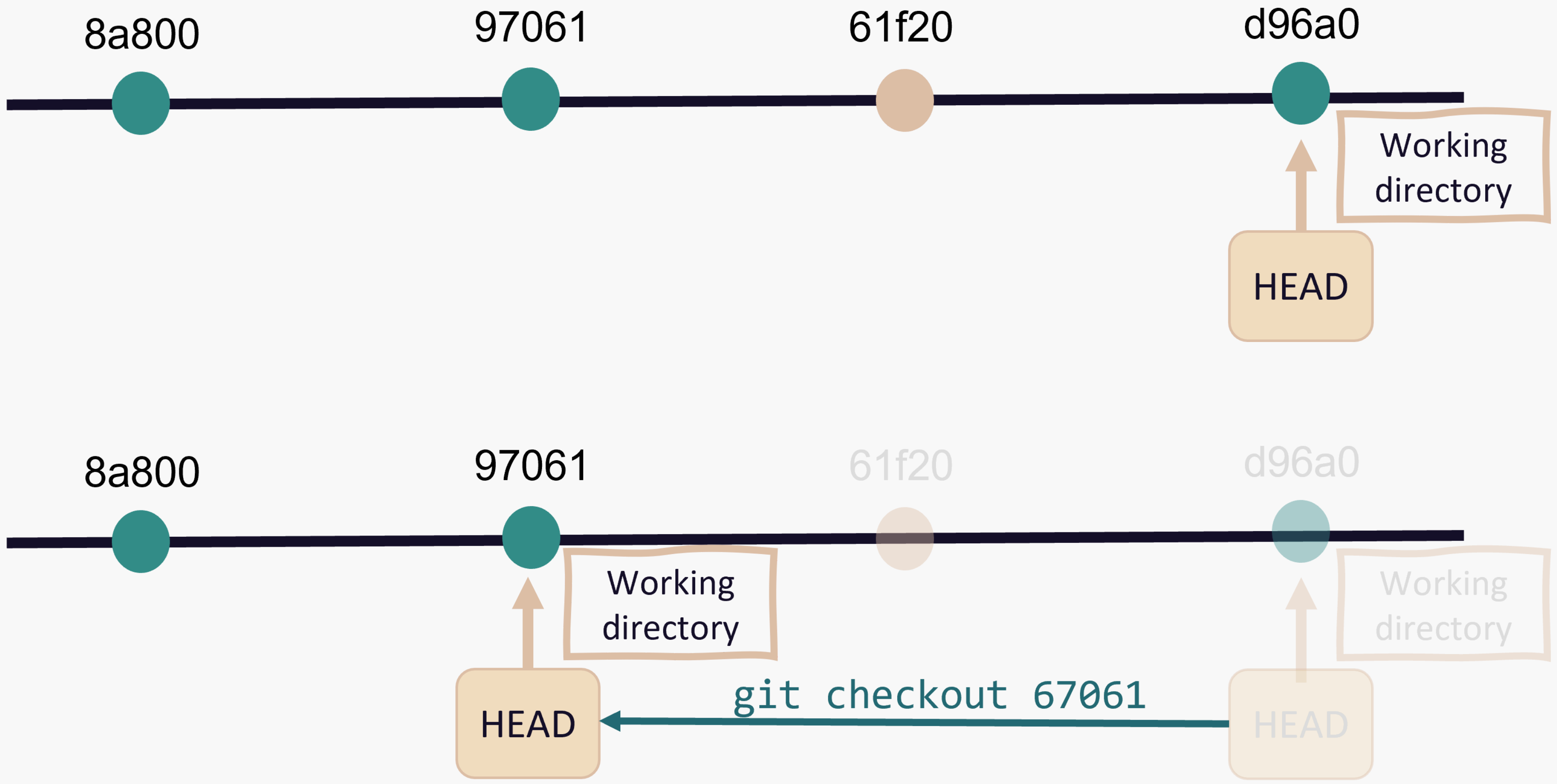
Additional topics

Undo accidental changes

- If you have non-commited changes
  - Discard your changes (right click on changes in GH Desktop > discard changes)
  - This also works for deleted files
- If you have commited changes
  - Use `git revert` OR
  - If you did not push yet undo the commit (right click on commit in GH Desktop > undo commit)
  - Then discard unwanted changes

Checkout a previous commit

Take your work space back in time temporarily with `git checkout`



Use tags

- Markers for specific commits (e.g. major milestones like “Submitted Manuscript” or “Version 1.0.”)
- Make it easy to return to a specific point in project’s history (e.g., if a reviewer requests changes based on an earlier analysis.)

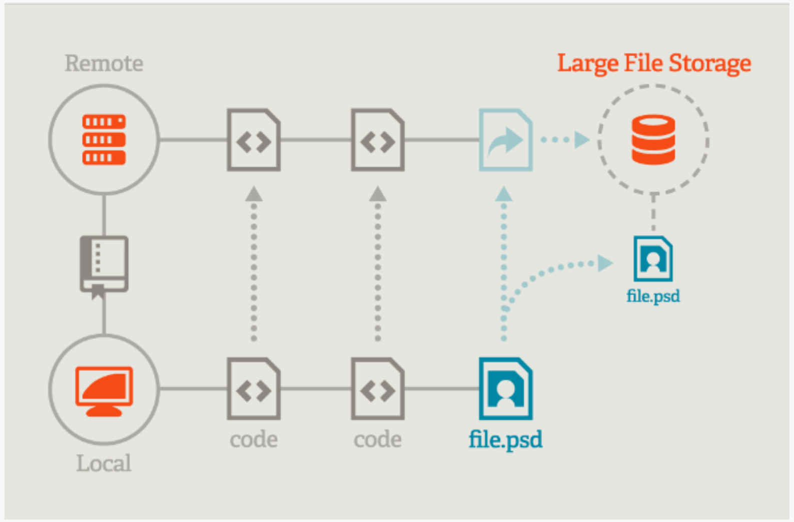
- How to Use Tags in GitHub Desktop:
  - Right-click on a commit in the history view.
  - Select “Create Tag.”
  - Name the tag (e.g., submission-v1, revision-accepted).

Releasing a clean version

- Before sharing your code, make sure it is clean and works.
- Remove temporary files, logs, and other files that are not needed.
- Use `.gitignore` to exclude files from version control.
- You can also create a branch with all extra files and then remove files from `main`
- Create a realeae on GitHub or even better, connect with Zenodo to get a DOI.

Git Large Fiels Storage (LFS)

- GitHub file size limit is 100 MB
- Git cannot track changes in binary files (e.g. `.png`, `docx`, ...)
  - every modification is treated as *completely new file*, leading to bloated history.
- Git LFS stores files outside the main history as lightweight pointers.
- Keeps repositories smaller and faster to clone.



Concept figure of Git LFS from https://git-lfs.com

Git Large Fiels Storage (LFS)

Use regular Git for:

- Text-based files (`.csv`, `.txt`, `.R`, `.md`, ...) with incremental changes

Use Git LFS for:

- large binary files (`.png`, `.pdf`, `.zip`, `.docx`, ...)
- Files > 50 MB that
  - change frequently
  - impacts the repository size

How to use Git LFS

1. **Download** from here
2. **Install**: Run `git lfs install` in the terminal.
3. **Track Files with LFS**: Use `git lfs track "*.png"` to configure tracking in `.gitattributes`.
4. **Commit**: Add `.gitattributes` and large files as usual.
5. **Push**: Upload changes with `git push`.

Licences

- If you publish a repository, you should add a license.
- For research output CC licenses are common.
  - Check out choosealicense.com for help.
  - Check out journal, project, university requirements.
- Add a licence using a `LICENSE` file
  - Copy the license text into a file and commit
  - Use GitHub's license generator: In the GitHub repo, click "Add File" > "Create New File" > Type LICENSE and select template.
  - In R use `usethis::use_*_license()` functions to add a license file.

Feedback

Please take 7 min to fill out the feedback form:

https://votingo.cedis.fu-berlin.de/survey/ARZC2L

- Use the free text questions in the end for suggestions

Questions?

How to continue from here

- Start using Git
- Have a look at additional resources
- Come to the tools and tips lecture for more science workflow topics
    - Starts again next summer semester
- Consultation: R programming, Git, …

Thanks for your participation

> See you in the next workshop 👋