Advanced cognitive neuroscience
Practical class: week 0

Mads Jensen

Fall, 2020

Git

Starting a git version control project
To start a git repository (repo), in a terminal go to the directory and type:

```
git init
```

This should produce output like: `Initialized empty Git repository in [directory name]`
This create a hidden folder named '.git' in the directory.

DO NOT MANUALLY TOUCH THIS FOLDER!

The command `git status` will print the *status* of the repo, i.e. are there new file added or changed etc.

Adding a file
Make a text file in the git repo directory, call it *file1.md*. To add the file to the repo type:

```
git add file1.md
```

Run git status again

Commit the file `git commit -m 'initial commit'`

Show the history of a repo
Often we are interested in seeing the history of a repo. The command:

```
git log
```

will show this.

tracking changes
Open the text file and change some of the text. If you run:

```
git diff
```

It should highlight the changes
Now add the file again and commit it. Look at the log.

Branching

One of git's best selling points is *branching*.
To view the branches in a repo:

```
git branch
```

To create a new branch:

```
git checkout -b fix
```

Change the file1.md, make a *new line* and enter more text. Add and commit the changes. Now checkout *master*

To get the changes from fix branch to master run:

```
git merge fix
```

Creating a merge conflict
Open *file1.md* edit the new line with some more text.
Checkout the *fix* branch and edit the same line as on master branch with some other text.
Checkout *master* again and try to merge.

To solve the merge conflict, open *file1.md* in an editor.

Undoing and previous versions of a file

Working with Remotes

For real collaborations you will need a shared place to host your code, this can be a local server or a server hosted in the cloud. In git terms we refer to such a place as a *remote*. For a good guide and introduction to working with remotes see the chapter in the Chacon & Straub (2014) called *working on remotes*.

The most used cloud platform for hosting git repositories are GitHub and I would recommend hosting your code there. It is possible to have the code repositories as *private*, meaning that only invited people have access to the code, just if you prefer that.

References

Chacon, S., & Straub, B. (2014). Pro git. Apress. https://git-scm.com/book/en/v2