Course Plan - Git & git.embl.de

Introduction

- Why worry about version control?
- "Poor man's Git" -> multiple folders/filenames with different versions of project/script
- What is git good for? What is Git *not* good for?

The GitHub Interface & First Steps

- log into GitHub
- show an example project
  - highlight key parts of project interface
  - show project history timeline
  - show blame for a single file
  - show how to view older versions of a file
- create a project from scratch - Guacamole
- add a README.md
- introduce basics of Markdown
- make a change to README via web interface
- commit and explain commit message
- repeat
- add new file(s)
  - instructions.txt:

```
* chop avocados
* chop onion
* and mix well
```

- ingredients.txt

```
* 2 avocados
* 1 lime
* 2 tsp salt
```

Working on the command line

- navigate to relevant directory
- clone repo && cd into project directory
- change global settings
  - `git config --global user.name`
  - `git config --global user.email`
  - `git config --global core.editor "nano -w"` (no wrapping)
  - `git config --global color.ui auto`
- `ls -a` -> `.git` folder
- mention `git init` - demo this at the end of day
- `git status`
- `git log`
- edit/create a file - add "1/2 onion" to ingredients
- `git add`
- `git commit`
  - (if necessary) more about commit messages
  - imagine your future self as a collaborator, who won't know (remember) why you made the changes you're making
- explain staging area; local repository
- EXERCISE -> slide
- one more change - "add salt to taste" to instructions
- `git add`
- `git reset HEAD instructions.txt`
- diagram on slide
- `git commit -m`
- mention `git commit -a` & warn about hazards of using it
- `git log`
  - `git log -N`
  - `git log --oneline`
  - `git log --patch <filename>`
- add "* tomato ketchup" to ingredients
- `git add`
- `git commit`
- "* squeeze lime" to instructions
- `git diff`
  - `git add + git diff --staged`
  - `git diff --color-words`
  - `git diff HEAD~2 <filename>`
  - `git diff <commithash> <filename>`
- `git revert <commithash>`
- edit README and another file
- `git checkout HEAD` to revert to most recent committed state
  - `git checkout HEAD <filename>` to achieve the same thing with a single file

- EXERCISE -> slide
- detached head!
  - `git checkout <commithash>` (forgetting filename)
  - `git checkout master` to recover from this
- `.gitignore`
- remotes
  - `git remote`
  - `git remote -v`
  - `git push origin master`
  - make a change on GitHub
  - `git pull origin master`
- collaboration exercises

Extras

- git init
- GitHub Flow
- issues, automatically closing
- setting up SSH keys