GETTING STARTED WITH VERSION CONTROL: INTRO TO GIT AND GITHUB

DATA CULTURE SOCIETY

Support for data-led and applied digital research across the arts, humanities and social sciences.

DATA CULTURE SOCIETY

Upcoming events & Training

- 16/11 Fika A chance to meet others working with digital resources and methods and talk tech over a cup of tea and a slice of cake
- 28/11 PhDs & ECR Social
- 29/11 Town Hall Meeting Autumn 2023
- 30/11 Masterclass on Keyphrase Network Analysis
- 01/12-08/12 Build your Personal or Project Website with GitHub Pages
- 13/12 Annual Lecture 23 Mary Flanagan “How to See What’s Missing”

THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society

DATA CULTURE SOCIETY

Welcome!

Sarah Schöttler

PhD student in data visualization & visualization developer

Aislinn Keogh

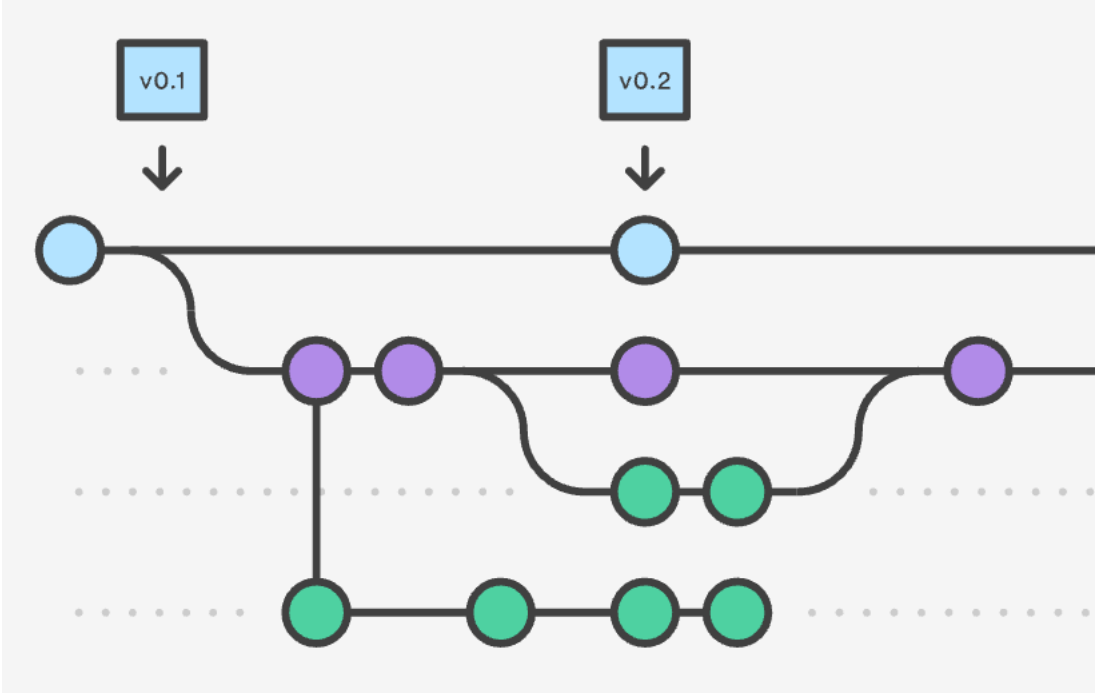
PhD student in Linguistics

Plan for Today

14:00	Welcome & Introduction
14:15	Live demo: Create your first repo & your first commit
	Troubleshooting
15:00	Break
15:10	Live demo: Fork a repo & create a pull request
	Troubleshooting
15:50	Wrap up & additional resources

Version Control

- **Version control is the practice of tracking and managing *changes* to software code.**
- Version control software keeps track of every modification to the code in a special kind of database.
- Version control helps tracking every individual change by each contributor and helping prevent ***concurrent*** work from conflicting.
- It is a standard ***workflow*** in the tech industry.



THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society

DATA CULTURE SOCIETY

git

A distributed version control system

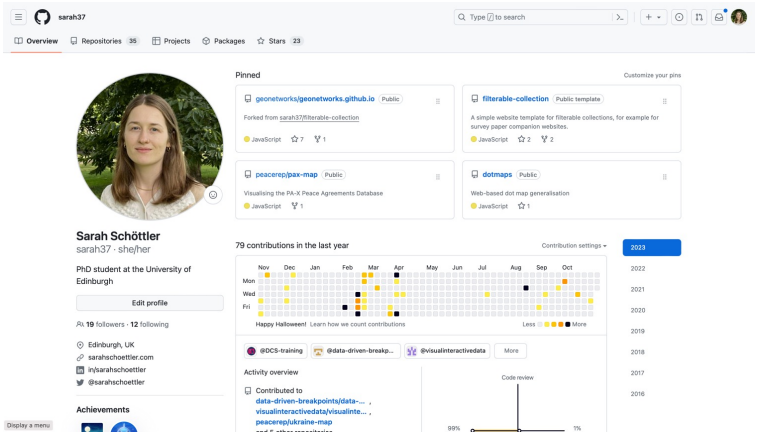
No need to learn this for basic use cases!

```
singh@DESKTOP-PGVSHMF MINGW64 ~/Desktop/newRepo (master)
$ git checkout master
Switched to branch 'master'

singh@DESKTOP-PGVSHMF MINGW64 ~/Desktop/newRepo (master)
$ git merge newbranch
Updating 9e7f7d0..3eb93e9
Fast-forward
 branch file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 branch file.txt

singh@DESKTOP-PGVSHMF MINGW64 ~/Desktop/newRepo (master)
$
```

Image from https://funkyvast.weebly.com/what-is-git-bash-terminal.html

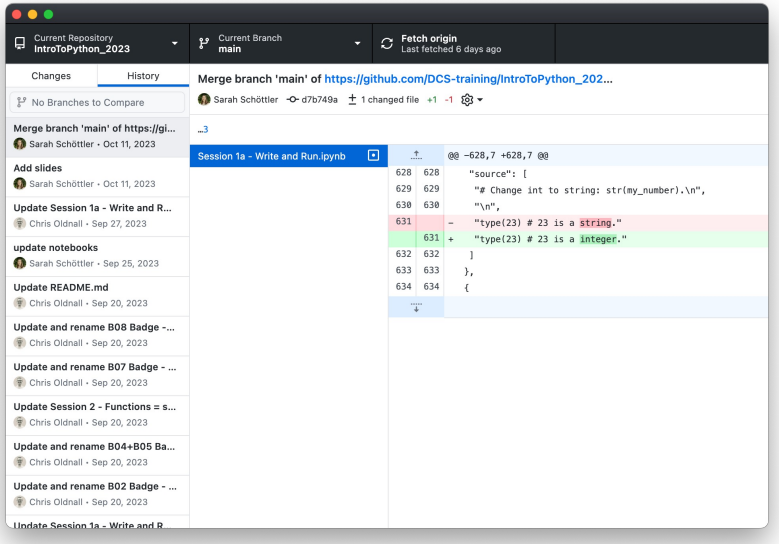
GitHub

An online platform for collaborating on code, based on git



GitHub Desktop

A desktop application for working with git and GitHub on your computer



Terminology

repository/repo

commit

clone

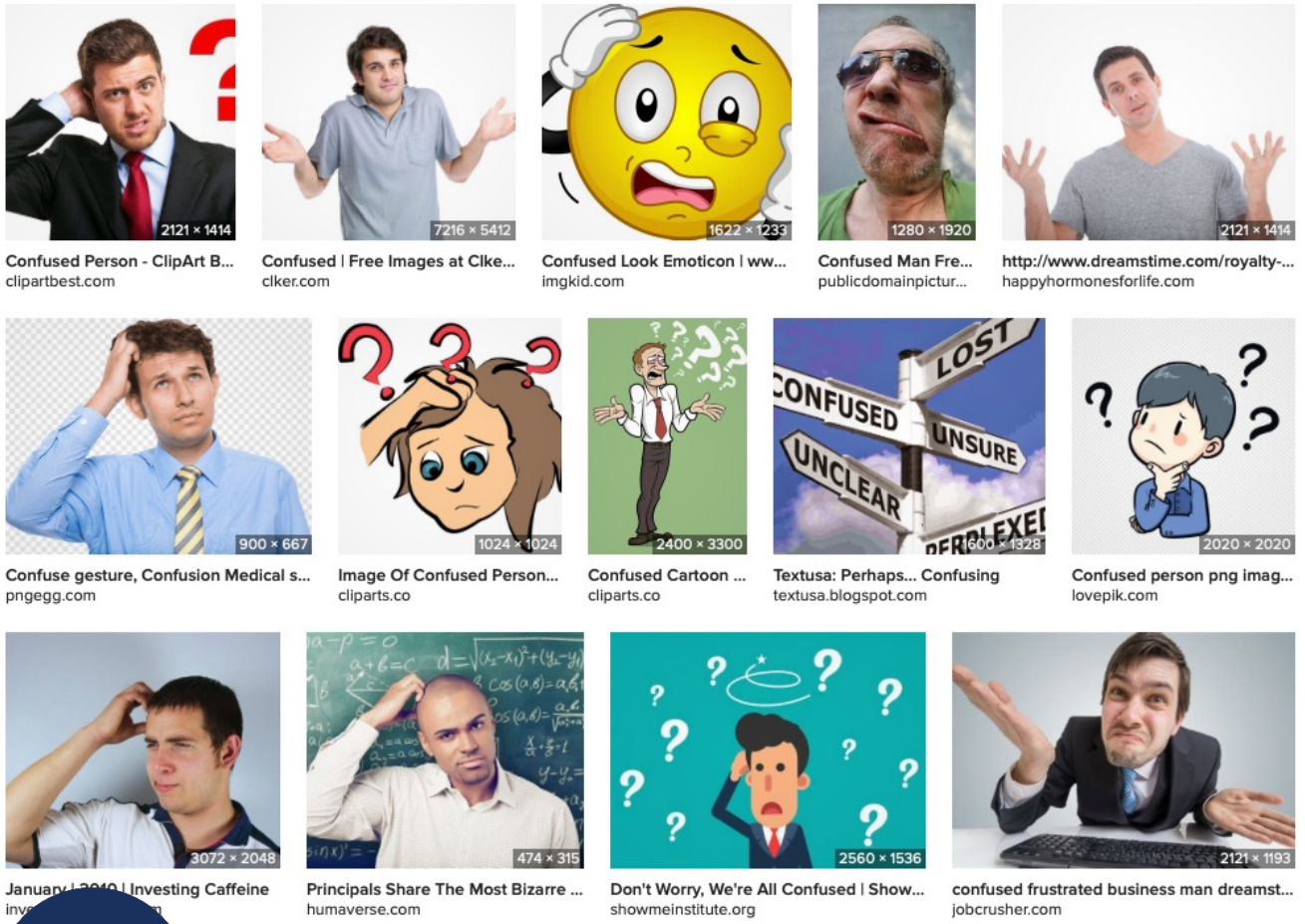
branch

fetch

pull

push

fork



Terminology

repository/repo → a “folder” for your code for one project

commit → saving your changes in the project

clone → a copy of a repo (e.g., on your computer)

branch

fetch

pull

push

fork

Live Demo:
Creating your first repo

1. Go to www.github.com
2. Sign in / sign up
3. Click the **+** in the top right corner and select “New repository”
4. Choose a repository name, check “Add a README file”, and leave all other fields.
5. Click “Create repository”

Live Demo:
Cloning your repo with GitHub Desktop

1. Install GitHub Desktop from https://desktop.github.com if you haven't yet
2. On the GitHub page for your repo, click "Open in GitHub Desktop"
3. Choose where to clone your repo
4. Click **fetch** & admire your repo in GitHub Desktop

Live Demo:
Making a commit

1. Open the README file in a text editor of your choice
   If you're unsure: on Windows use **Notepad**; on Mac use **TextEdit**
2. Make some changes in your README file and save them
3. In GitHub Desktop, describe your commit and click commit when ready
4. Push to GitHub

Break


GETTING STARTED WITH
VERSION CONTROL: INTRO
INTRO TO GIT AND GITHUB

Terminology

repository/repo	→ a “folder” for your code for one project
commit	→ saving your changes in the project
clone	→ a copy of a repo (e.g., on your computer)
branch	→ a parallel version of a repo (repos can have multiple branches)
fetch	→ getting the latest changes from the online repo
pull	→ integrating the latest changes into your clone of the repo
push	→ sending your changes (your commits) to the server
fork	→ a personal copy of someone else’s repo

Live Demo:
Forking a repo, making a commit, and making a pull request

1. Find a partner and get the link to their GitHub repo. Fork it on GitHub, then add your fork to GitHub Desktop.
2. Make a change to their README, commit, and push.
3. On GitHub, view your fork and open a pull request.
4. Review and approve each other's pull requests.

Additional Resources

- GitHub online tutorials
  - Introduction (tutorial repeating some of what we did today): https://github.com/skills/introduction-to-github
  - Reviewing pull requests: https://github.com/skills/review-pull-requests
  - Resolving merge conflicts: https://github.com/skills/resolve-merge-conflicts
  - List of all skills tutorials: https://github.com/skills
- GitHub student pack: https://education.github.com/pack