Essential Research Toolkit for the Humanities

Week 14: Git(Hub) basics

Anna Pryslopska

July 11, 2022

Psycholinguistics and Cognitive Modeling Lab

Homework

Use the terminal not R.

Find the changes I made in the big project
Unpack files and compare them in pairs
Look also in subfolders

How many times does the word “verhalten” appear in the corpus files?

```
grep "verhalten" -c corpus1.TXT corpus2.TXT corpus3.TXT
corpus1.TXT:90
corpus2.TXT:145
corpus3.TXT:146
```

`-i` = ignore case
`-o` = (print) only matching

Homework

How many times (not where/list of cases) does “Berlin” appear in corpus files? BeRlIn ✔, Berliner ✖

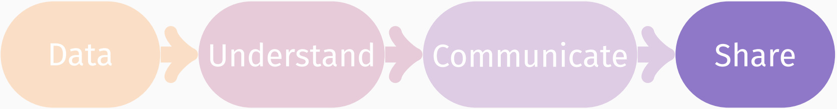
```
grep -w -c -i "Berlin" corpus1.TXT corpus2.TXT corpus3.TXT
corpus1.TXT:8
corpus2.TXT:4
corpus3.TXT:4
```

What are the differences between `corpus2.txt` and `corpus3.txt`?
`diff corpus2.txt corpus3.txt` (no quotation marks) 6 changes

`telnet ateraan.com 4002` → command not found and other issues

Questions?

Sharing



Data

Understand

Communicate

Share

import, clean, transform, export, visualize, model

document, create clean and beautiful reports

connect, collaborate, backup

Table of contents

1. Version control

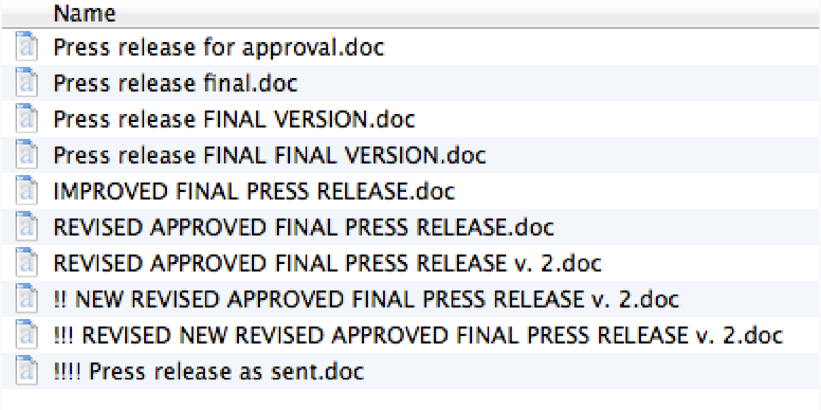
2. Git and GitHub

3. Getting started

4. Homework assignment

Version control

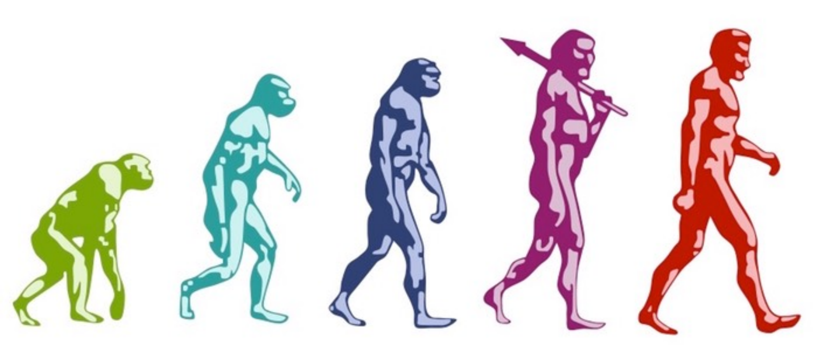
Does this look familiar?



Name
Press release for approval.doc
Press release final.doc
Press release FINAL VERSION.doc
Press release FINAL FINAL VERSION.doc
IMPROVED FINAL PRESS RELEASE.doc
REVISED APPROVED FINAL PRESS RELEASE.doc
REVISED APPROVED FINAL PRESS RELEASE v. 2.doc
!! NEW REVISED APPROVED FINAL PRESS RELEASE v. 2.doc
!!! REVISED NEW REVISED APPROVED FINAL PRESS RELEASE v. 2.doc
!!!! Press release as sent.doc

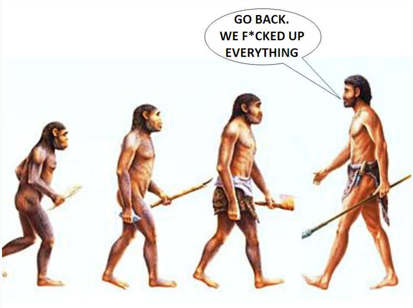
Popular choices include: date, name, version nr, final, copy, reviewed.

Version control



Version control helps you track changes over time.

Version control



When you mess up you can go back to when things worked.

Renaming, saving as, adding dates and initials etc. to file names may work for a small project but breaks down fast with big projects.

Version control is commonly used in (software) development, in fast-paced work environments, and big teams.

Allows you to save stages of your project.

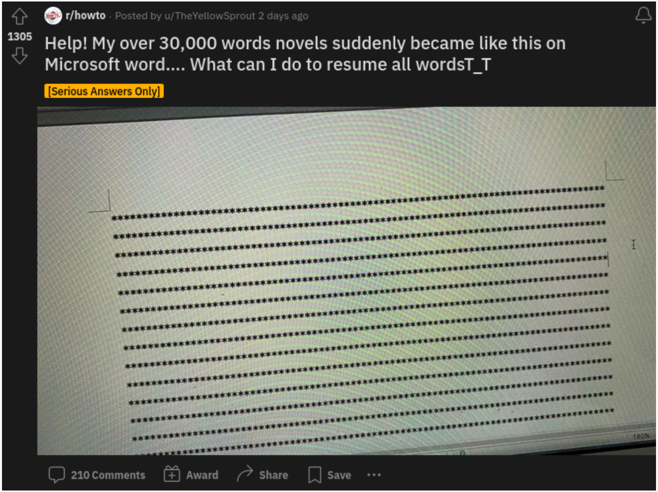
Tracking changes

`diff` is great locally and you can use DiffChecker `https://www.diffchecker.com/` → paid + online

Multiple people working on the same project in parallel makes files messy, fast:

- Collaborator 1 is writing the intro
- Collaborator 2 is adding the statistics
- Collaborator 3 is summarizing the results
- Proofreader is checking for errors
- ...

Backup



r/howto · Posted by u/TheYellowSprout 2 days ago

1305

Help! My over 30,000 words novels suddenly became like this on Microsoft word.... What can I do to resume all wordsT_T

[Serious Answers Only]

Remember to backup important files locally and in the cloud!

Git and GitHub

Why learn Git?

- Keep track of your data
- Synchronize across devices
- Collaborate effectively
- Resolve errors and fix mistakes
- Parallelize processes
- Backup and store files
- Remember, revert, and restore (short- and long-term)
- Be more employable as (data) scientists and developers
- It’s free and open source

Git

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

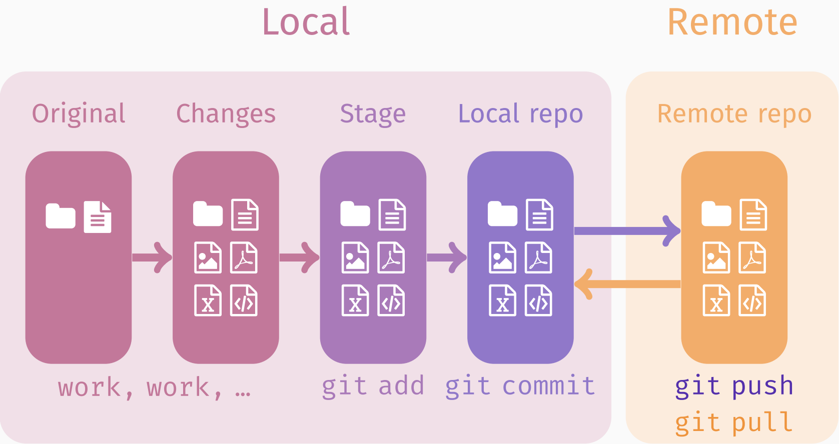
Git, GitHub, and GitLab

Git (🇬🇧 slang: worthless, foolish, unpleasant person) is an OS version control software that works locally on your computer. Great for individual use.

GitHub is the most popular code hosting platform that makes collaborating using git easier. LinkedIn and Twitter for developers, owned by Microsoft. Uni Stuttgart has their own GitHub server `https://github.tik.uni-stuttgart.de/`

GitLab is an alternative code hosting platform that has somewhat different user permissions and includes Continuous Integration (no need to pull) → this will be explained soon.

Workflow



Local
Remote
Original
Changes
Stage
Local repo
Remote repo
work, work, …
git add
git commit
git push
git pull

Glossary: Setup

Repository (repo) a place where something (e.g. data) is stored and managed.

Server the computer storing the repository.

Client the computer connecting to the repository.

Working set/copy local directory/place where you make changes.

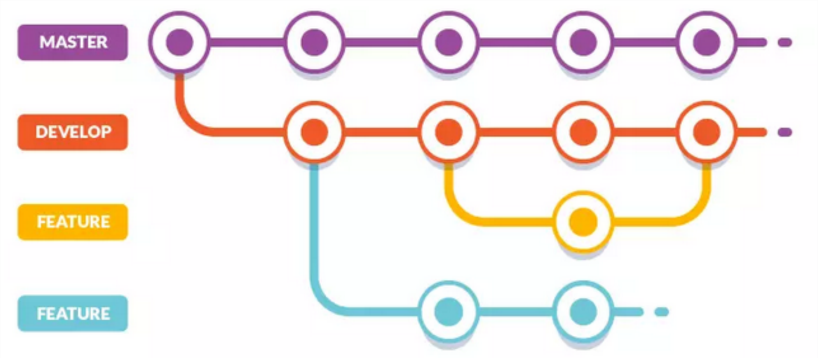
Trunk/main/master the main location for the data in the repository.

Fork is a copy of a repository that you can change to your liking.

Branch a parallel version of the repository at this time.

Merge combine a (feature) branch with the main branch/trunk.

Tree structure



Source: Sitepoint

Glossary: Actions (must know)

Add tell git to track untracked files.

Commit change to tracked files. Has a unique ID ($\approx$ save as) and keeps record of changes. Usually contains a message, i.e. description of changes.

Status what are the changes since the last commit?

Push send changes to a remote repository.

Pull get changes from a remote repository.

Glossary: Actions

Changelog list of all changes.

Branch (make a) parallel version of a repository.

Diff difference in changes between files.

Merge combining changes from two branches.

Reset undo/rollback the last commit (move backwards).

Revert undo/cancel a change but make a new commit (move forwards).

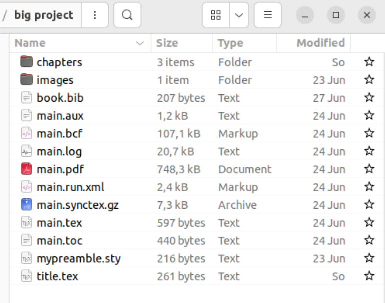
Getting started

Install git and set it up

1. Download git e.g.
   `https://github.com/git-guides/install-git`
2. Open a terminal
3. Give yourself a name:
   ```
   git config --global user.name "Your Name"
   ```
   - which program
   - (topic) do what and where
   - (variable) details please
   - your name or alias
4. Insert your email address:
   ```
   git config --global user.email "mail@example.com"
   ```

First repo

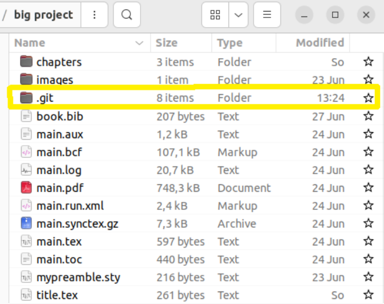
Navigate via command line (`cd`) or open in terminal.



Name	Size	Type	Modified	
chapters	3 items	Folder	So	☆
images	1 item	Folder	23 Jun	☆
book.bib	207 bytes	Text	27 Jun	☆
main.aux	1,2 kB	Text	24 Jun	☆
main.bcf	107,1 kB	Markup	24 Jun	☆
main.log	20,7 kB	Text	24 Jun	☆
main.pdf	748,3 kB	Document	24 Jun	☆
main.run.xml	2,4 kB	Markup	24 Jun	☆
main.synctex.gz	7,3 kB	Archive	24 Jun	☆
main.tex	597 bytes	Text	24 Jun	☆
main.toc	440 bytes	Text	24 Jun	☆
mypreamble.sty	216 bytes	Text	23 Jun	☆
title.tex	261 bytes	Text	So	☆

Initialize git:

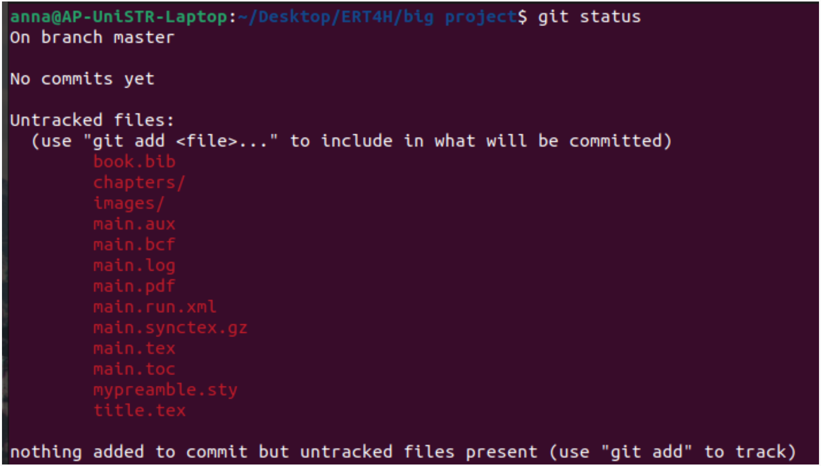
```
git init
```



Name	Size	Type	Modified	
chapters	3 items	Folder	So	☆
images	1 item	Folder	23 Jun	☆
.git	8 items	Folder	13:24	☆
book.bib	207 bytes	Text	27 Jun	☆
main.aux	1,2 kB	Text	24 Jun	☆
main.bcf	107,1 kB	Markup	24 Jun	☆
main.log	20,7 kB	Text	24 Jun	☆
main.pdf	748,3 kB	Document	24 Jun	☆
main.run.xml	2,4 kB	Markup	24 Jun	☆
main.synctex.gz	7,3 kB	Archive	24 Jun	☆
main.tex	597 bytes	Text	24 Jun	☆
main.toc	440 bytes	Text	24 Jun	☆
mypreamble.sty	216 bytes	Text	23 Jun	☆
title.tex	261 bytes	Text	So	☆

What's going on

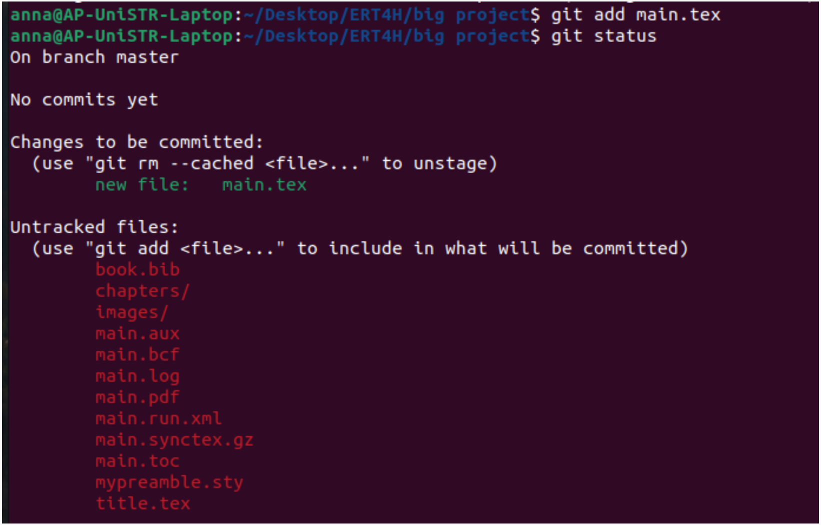
`git status`



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        book.bib
        chapters/
        images/
        main.aux
        main.bcf
        main.log
        main.pdf
        main.run.xml
        main.synctex.gz
        main.tex
        main.toc
        mypreamble.sty
        title.tex

nothing added to commit but untracked files present (use "git add" to track)
```

What to track

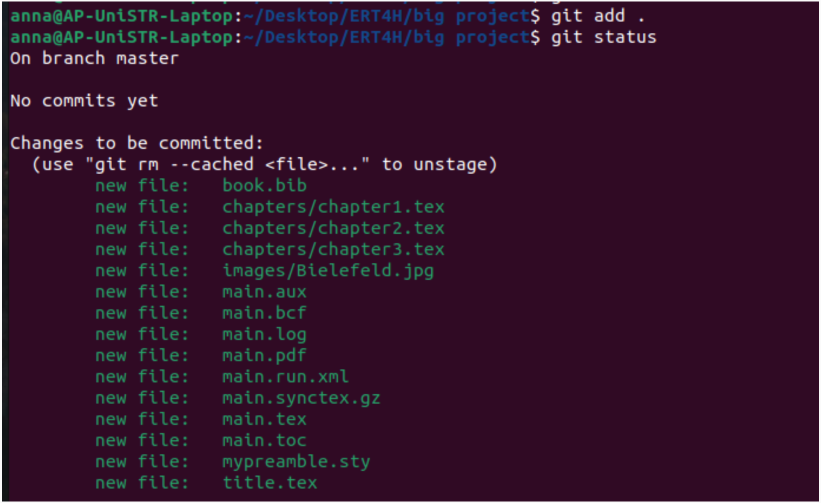
git add FILE



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git add main.tex
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   main.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        book.bib
        chapters/
        images/
        main.aux
        main.bcf
        main.log
        main.pdf
        main.run.xml
        main.synctex.gz
        main.toc
        mypreamble.sty
        title.tex
```

Keep all records

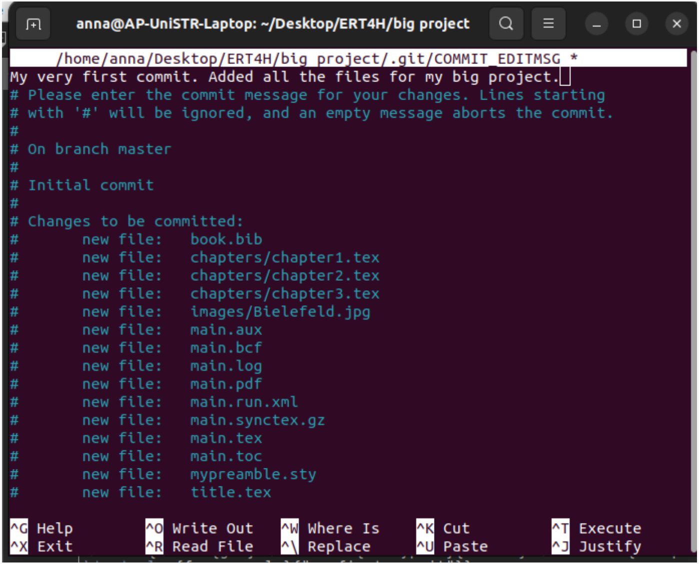
```
git add .
```



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git add .
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   book.bib
        new file:   chapters/chapter1.tex
        new file:   chapters/chapter2.tex
        new file:   chapters/chapter3.tex
        new file:   images/Bielefeld.jpg
        new file:   main.aux
        new file:   main.bcf
        new file:   main.log
        new file:   main.pdf
        new file:   main.run.xml
        new file:   main.synctex.gz
        new file:   main.tex
        new file:   main.toc
        new file:   mypreamble.sty
        new file:   title.tex
```

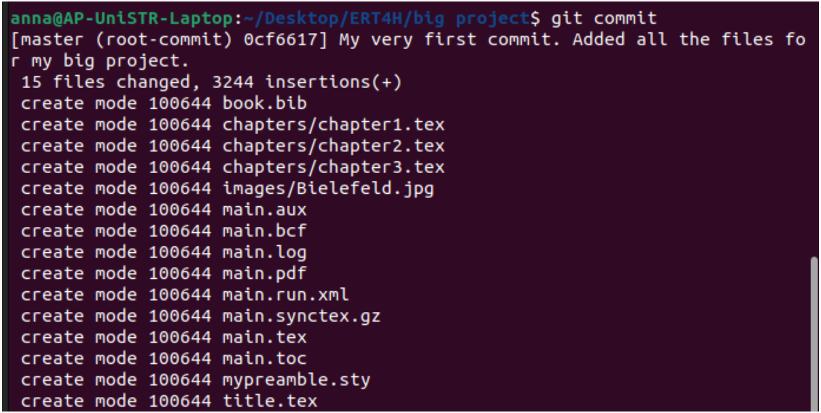
What have I done?

`git commit`



```
     /home/anna/Desktop/ERT4H/big project/.git/COMMIT_EDITMSG *
My very first commit. Added all the files for my big project.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   book.bib
#       new file:   chapters/chapter1.tex
#       new file:   chapters/chapter2.tex
#       new file:   chapters/chapter3.tex
#       new file:   images/Bielefeld.jpg
#       new file:   main.aux
#       new file:   main.bcf
#       new file:   main.log
#       new file:   main.pdf
#       new file:   main.run.xml
#       new file:   main.synctex.gz
#       new file:   main.tex
#       new file:   main.toc
#       new file:   mypreamble.sty
#       new file:   title.tex

^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify
```

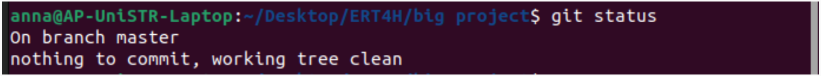
Changes



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git commit
[master (root-commit) 0cf6617] My very first commit. Added all the files fo
r my big project.
 15 files changed, 3244 insertions(+)
 create mode 100644 book.bib
 create mode 100644 chapters/chapter1.tex
 create mode 100644 chapters/chapter2.tex
 create mode 100644 chapters/chapter3.tex
 create mode 100644 images/Bielefeld.jpg
 create mode 100644 main.aux
 create mode 100644 main.bcf
 create mode 100644 main.log
 create mode 100644 main.pdf
 create mode 100644 main.run.xml
 create mode 100644 main.synctex.gz
 create mode 100644 main.tex
 create mode 100644 main.toc
 create mode 100644 mypreamble.sty
 create mode 100644 title.tex
```

```
git commit -m "My very first commit. Added all the
files for my big project."
```

What’s going on

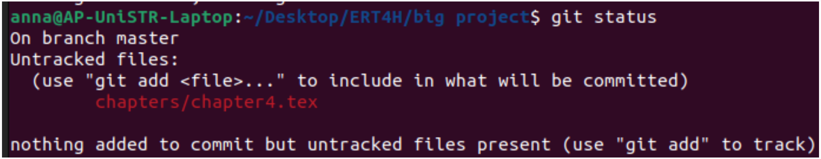


```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git status
On branch master
nothing to commit, working tree clean
```

MUCH
LATER

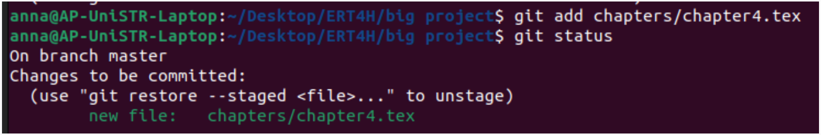
Round 2. Fight!

git status



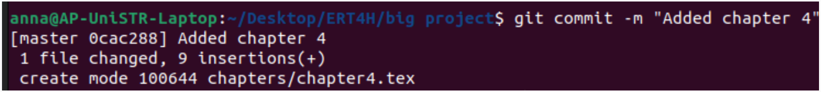
```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        chapters/chapter4.tex

nothing added to commit but untracked files present (use "git add" to track)
```

git add .



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git add chapters/chapter4.tex
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   chapters/chapter4.tex
```

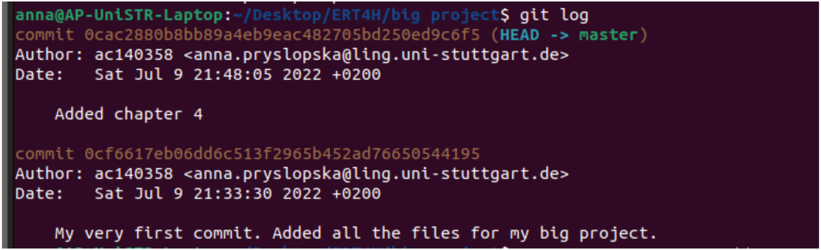
git commit



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git commit -m "Added chapter 4"
[master 0cac288] Added chapter 4
 1 file changed, 9 insertions(+)
 create mode 100644 chapters/chapter4.tex
```

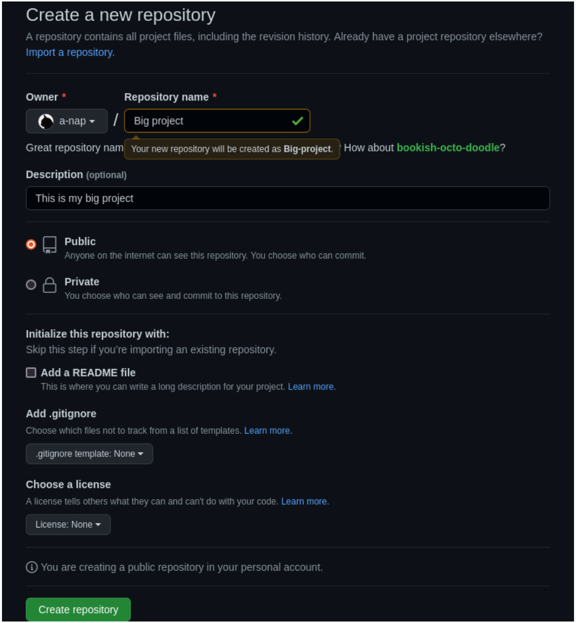
Captain’s log. Stardate 202207.11

`git log`



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git log
commit 0cac2880b8bb89a4eb9eac482705bd250ed9c6f5 (HEAD -> master)
Author: ac140358 <anna.pryslopska@ling.uni-stuttgart.de>
Date:   Sat Jul 9 21:48:05 2022 +0200

    Added chapter 4

commit 0cf6617eb06dd6c513f2965b452ad76650544195
Author: ac140358 <anna.pryslopska@ling.uni-stuttgart.de>
Date:   Sat Jul 9 21:33:30 2022 +0200

    My very first commit. Added all the files for my big project.
```

Taking it off the ground



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * a-nap / Repository name * Big project ✓

Great repository nam Your new repository will be created as **Big-project**. How about **bookish-octo-doodle**?

Description (optional)

This is my big project

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ **Add a README file**
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.

License: None

ⓘ You are creating a public repository in your personal account.

Pre-boarding

Passanger/owner name (e.g. ”GitHub Username” = ”e.shellstrop”):

```
git config --global user.username "GitHub Username"
```

Add your remote repository (e.g. URL = `https://github.com/e.shellstrop/shrimp`):

```
git remote add origin URL
```

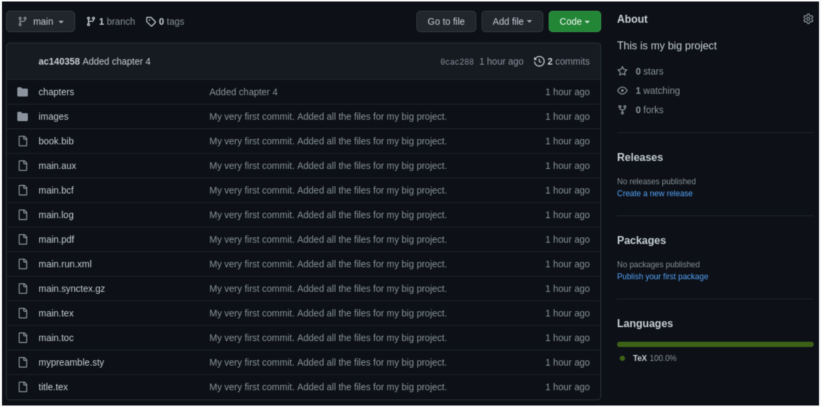
Rename branch to something a bit less racist:

```
git branch -M main
```

Send it on its way:

```
git push -u origin main
```

You did it!



ac140358 Added chapter 4 0cac288 1 hour ago **2** commits

chapters	Added chapter 4	1 hour ago
images	My very first commit. Added all the files for my big project.	1 hour ago
book.bib	My very first commit. Added all the files for my big project.	1 hour ago
main.aux	My very first commit. Added all the files for my big project.	1 hour ago
main.bcf	My very first commit. Added all the files for my big project.	1 hour ago
main.log	My very first commit. Added all the files for my big project.	1 hour ago
main.pdf	My very first commit. Added all the files for my big project.	1 hour ago
main.run.xml	My very first commit. Added all the files for my big project.	1 hour ago
main.synctex.gz	My very first commit. Added all the files for my big project.	1 hour ago
main.tex	My very first commit. Added all the files for my big project.	1 hour ago
main.toc	My very first commit. Added all the files for my big project.	1 hour ago
mypreamble.sty	My very first commit. Added all the files for my big project.	1 hour ago
title.tex	My very first commit. Added all the files for my big project.	1 hour ago

About

This is my big project

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

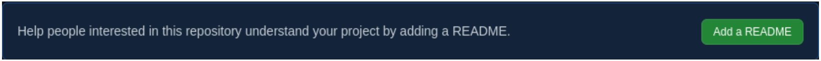
● **TeX** 100.0%

Staging

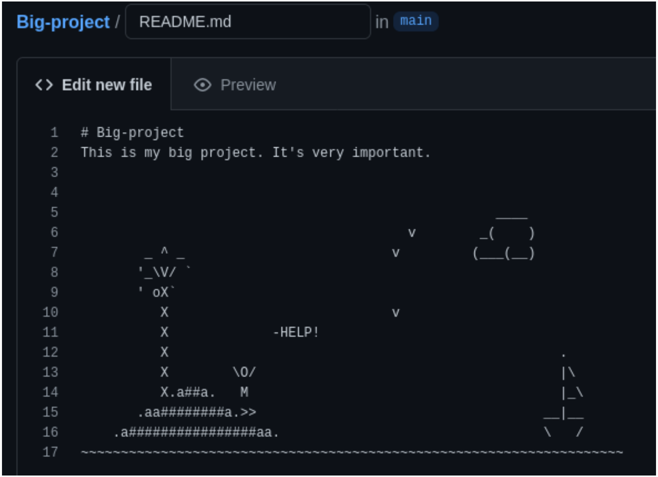
git commit
AIRBUS A350-1000
1000
A350
git push
AIRBUS A350-1000
1000
A350
git add .

Remote changes

Add some information about your project



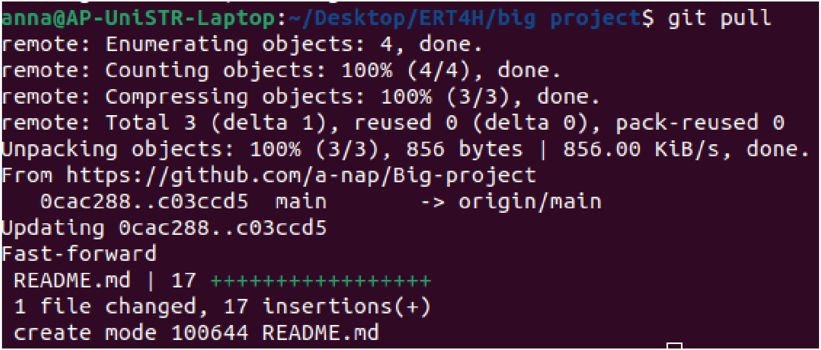
Help people interested in this repository understand your project by adding a README.



```
# Big-project
This is my big project. It's very important.



                                                    ____
                                         v        _(    )
        _ ^ _                          v         (___(__)
       '_\V/ `
       ' oX`
          X                            v
          X             -HELP!
          X                                                 .
          X        \O/                                      |\
          X.a##a.   M                                       |_\
       .aa########a.>>                                    __|__
    .a################aa.                                 \   /
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Pulling your weight

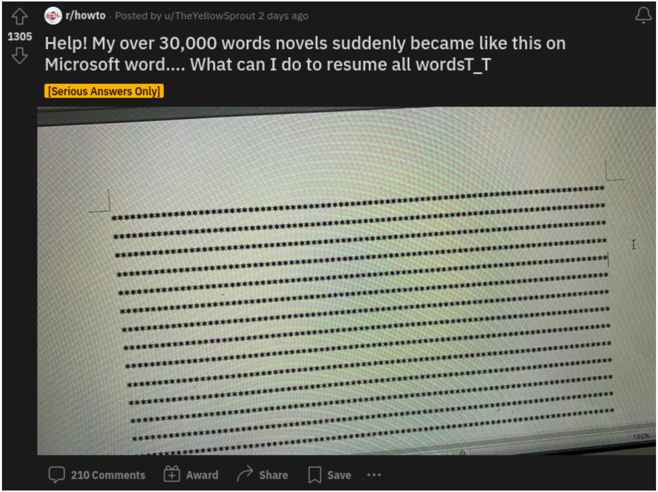
```
git pull
```



```
anna@AP-UniSTR-Laptop:~/Desktop/ERT4H/big project$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 856 bytes | 856.00 KiB/s, done.
From https://github.com/a-nap/Big-project
   0cac288..c03ccd5  main       -> origin/main
Updating 0cac288..c03ccd5
Fast-forward
 README.md | 17 +++++++++++++++++
 1 file changed, 17 insertions(+)
 create mode 100644 README.md
```

It may be confusing at first...

...but it's worth it.



r/howto · Posted by u/TheYellowSprout 2 days ago

1305

Help! My over 30,000 words novels suddenly became like this on Microsoft word.... What can I do to resume all wordsT_T

[Serious Answers Only]

Read more

Git guide: `https://github.com/git-guides/`

Another git guide: `http://rogerdudler.github.io/git-guide/`

Git tutorial: `http://git-scm.com/docs/gittutorial`

Another git tutorial: `https://www.w3schools.com/git/`

Git cheat sheets: `https://training.github.com/`

Ask questions: `https://stackoverflow.com`

Homework assignment

Homework assignment due July 16

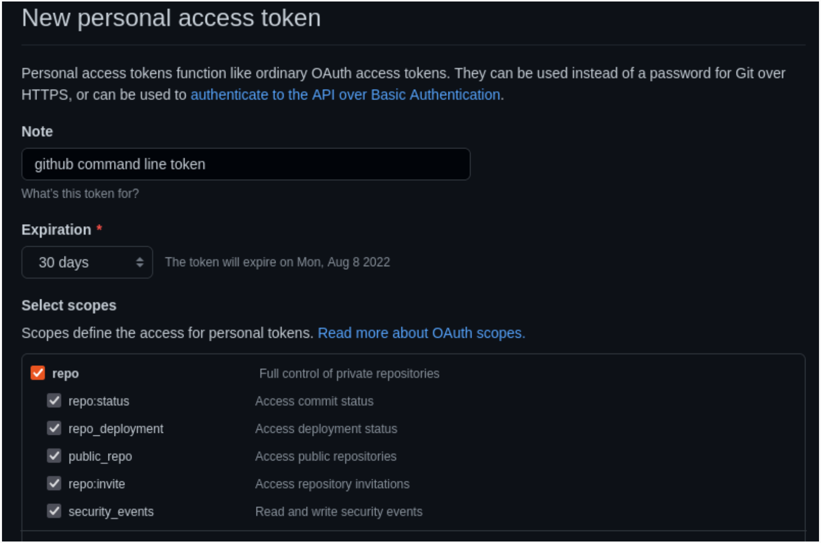
Complete the Git It tutorial:
`https://github.com/jlord/git-it-electron/releases`

Send me a link to your GitHub repository.

Earlier deadline because I will need time and internet to check it

Troubleshooting the Git It tutorial:

- For solutions and known issues, see `https://github.com/jlord/git-it-electron/issues`
- Can't log in? Did you set `git config --global user.username GITHUB USERNAME` correctly?
- Still can't log in? You might need to log in using NOT a password but a **personal access token**. To make one:
  GitHub Account Settings → Developer settings → Personal access tokens → Generate new token → Generate token



New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

github command line token

What's this token for?

Expiration *

30 days

The token will expire on Mon, Aug 8 2022

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

☑ repo	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events

Save it somewhere safe and use as a password for logging in via terminal.