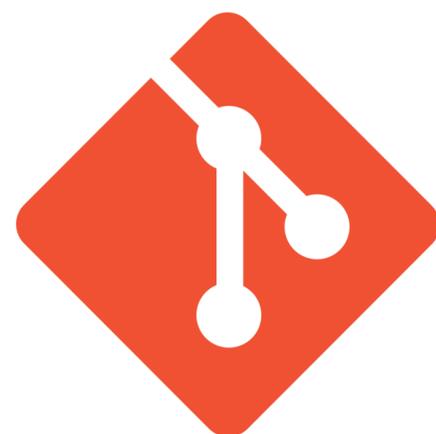


Praxislabor Digitale Geisteswissenschaften

Einführung in Git



git

Von Jason Long, CC BY 3.0

CC-BY 4.0

Gerrit Heim

g.heim@ub.uni-frankfurt.de

Rückblick

Was haben wir gemacht?

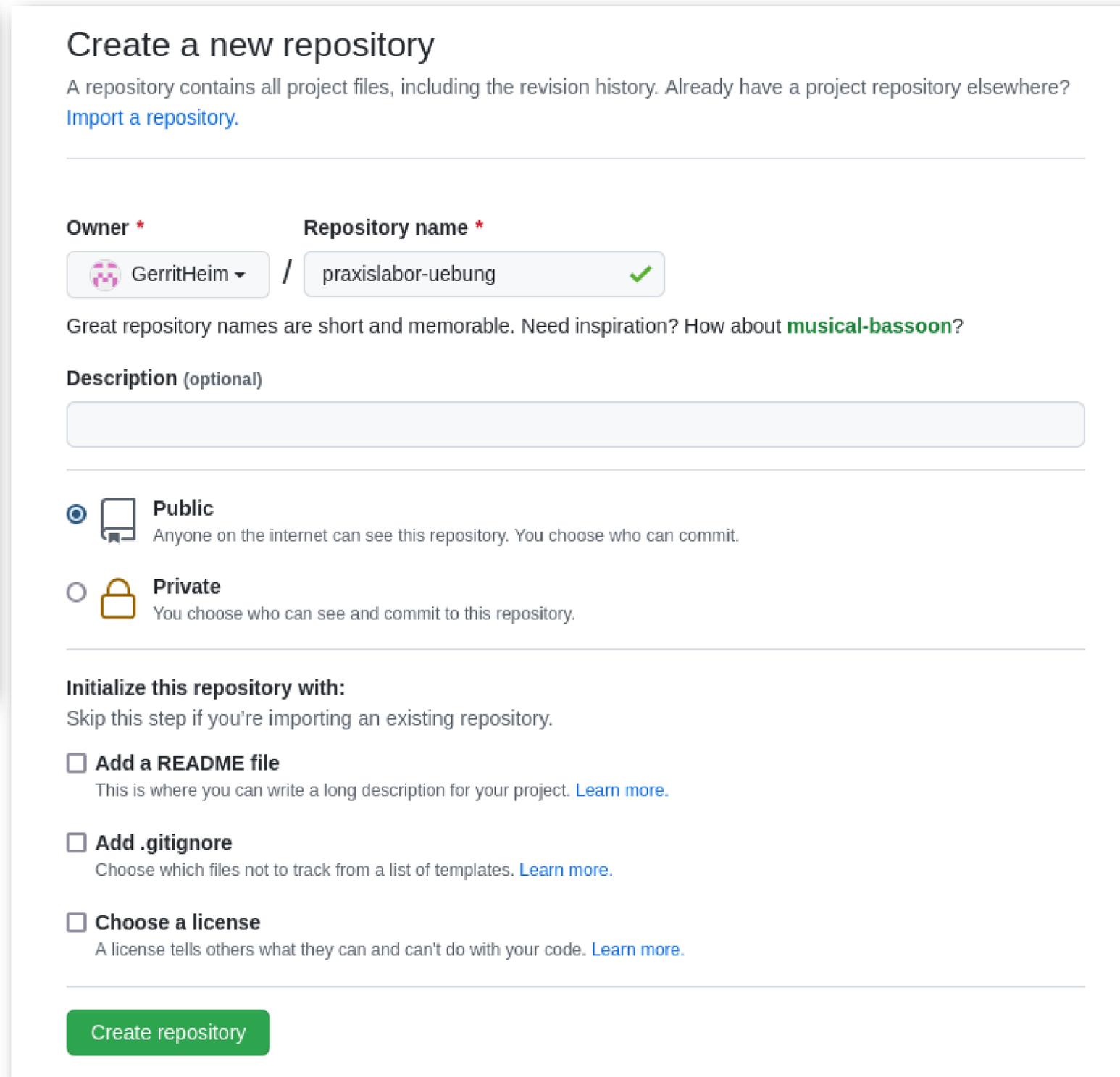
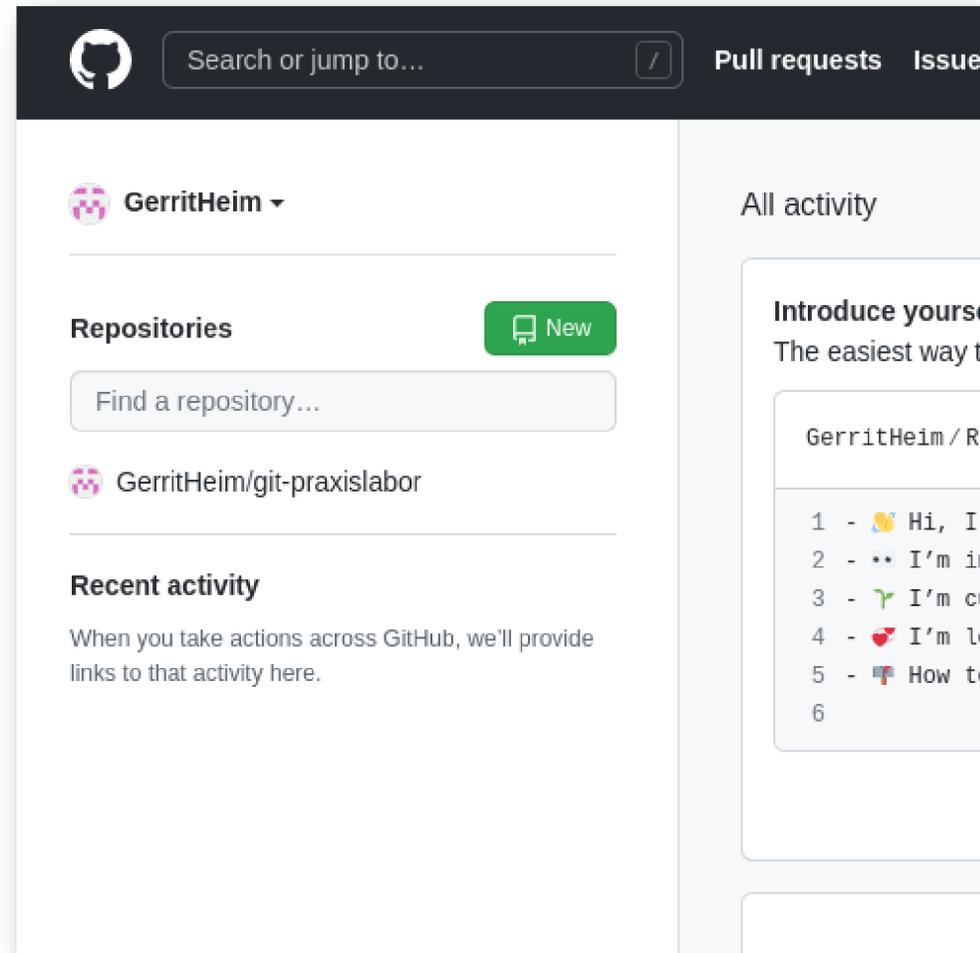
- Git konfiguriert
- Ein lokales Repository angelegt
- Dateien hinzugefügt und bearbeitet
- Commits erstellt
- Versionen zurückgesetzt



- GitHub Repository erstellen
- Lokales Git Repository mit GitHub Repository verknüpfen
- Kollaboratives Arbeit mit GitHub
- Grafische Software für Git



Lokales Repository mit GitHub verbinden 1/3



Lokales Repository mit GitHub verbinden 2/3

\$ git remote add origin <https://github.com/<username>/praxislabor-uebung.git>

Fügt das entfernte Repository hinzu. Dabei ist <username> durch den GitHub-Nutzer zu ersetzen.

\$ git remote -v

Zeigt die verknüpften entfernten Repositorien

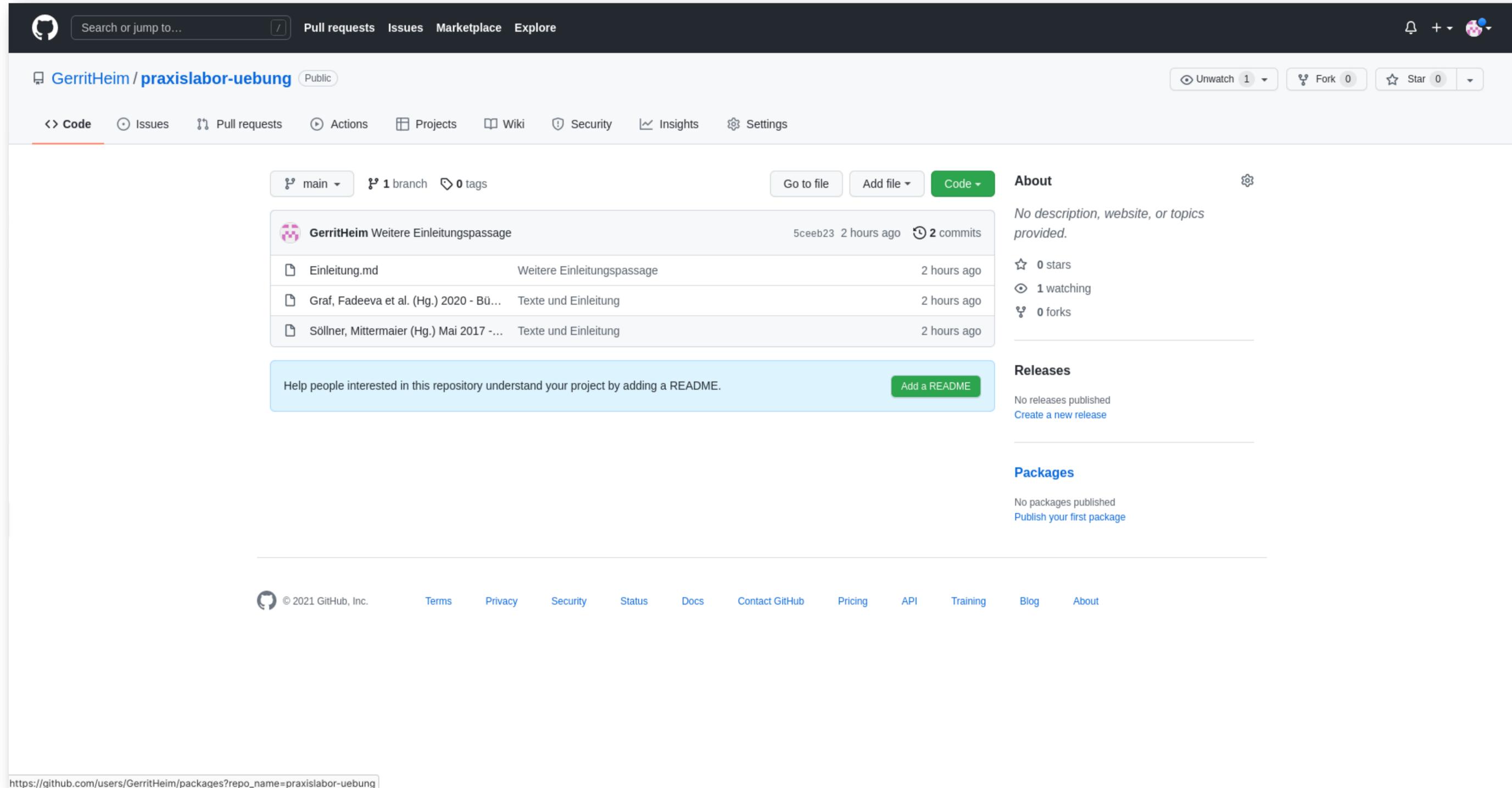
\$ git branch -M main

\$ git push -u origin main

Kopiert die bisherigen lokalen Dateien zu GitHub. Der Zusatz -u sorgt dafür, dass Git sich das merkt und für folgende Änderung \$ git push genügt.



Lokales Repository mit GitHub verbinden 3/3



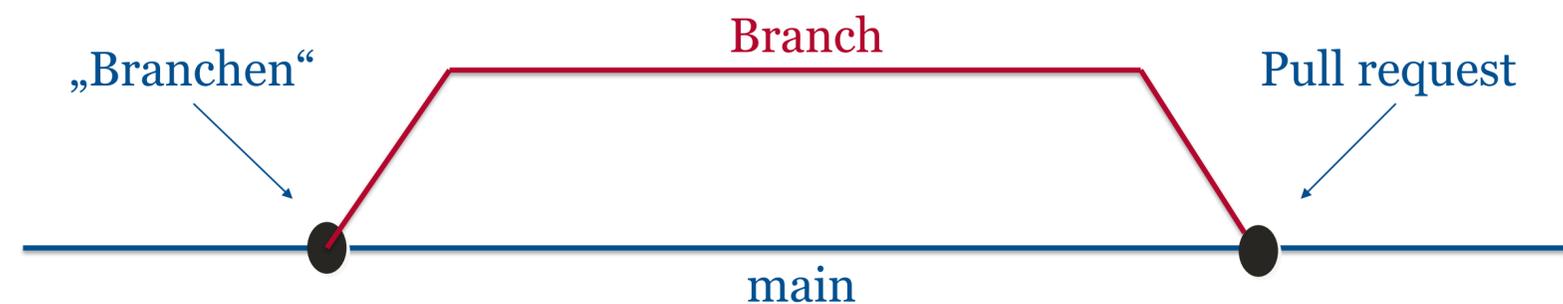
The screenshot shows the GitHub interface for a repository named 'praxislabor-uebung' by user 'GerritHeim'. The repository is public and has 1 branch (main) and 0 tags. It contains three files: 'Einleitung.md', 'Graf, Fadeeva et al. (Hg.) 2020 - Bü...', and 'Söllner, Mittermaier (Hg.) Mai 2017 - ...'. The repository has 0 stars, 1 watcher, and 0 forks. There are no releases or packages published. The footer of the page includes the GitHub logo, copyright information for 2021, and various links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.



https://github.com/users/GerritHeim/packages?repo_name=praxislabor-uebung

Kollaboratives Arbeiten: Theorie

- Bei kollaborativem Arbeiten existiert bereits meist ein Repository mit dem aktuellen Arbeitsstand des Projekts.
- Um Konflikte zu verhindern legt man einen separaten Arbeitszweig an („branchen“)
- Dort kann man unabhängig voneinander an Dateien arbeiten und diese am Schluss wieder zusammen führen („mergen“)



Kollaboratives Arbeiten: Praxis mit GitHub

```
$ cd ~/Desktop
```

Navigation an den Ort, wo das Repository auf dem System gespeichert werden soll.

```
$ git clone https://github.com/GerritHeim/praxislabor-uebung2.git
```

Beziehen der Daten aus dem GitHub Repository

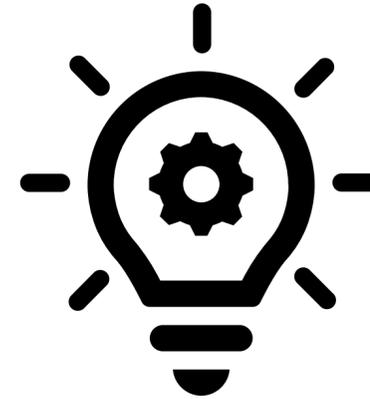
```
$ git checkout -b <Branchname>
```

Erstellt einen neuen Branch mit dem vergeben Namen und wechselt in diesen Branch.



Kollaboratives Arbeiten: 4 Grundschritte

1. Dateien erstellen / bearbeiten
2. Änderungen betrachten
3. Änderungen für Abgabe vorbereiten (Staging)
4. Änderungen abgeben (Commiten)



Beispiel:

Datei erstellen und mit Inhalt füllen

Zusammenfassung überlegen

```
$ git add GerritHeim.md
```

```
$ git commit -m „Datei von Gerrit Heim“
```

Zum Schluss den Inhalt auf GitHub übertragen (hierfür muss aber ihr GitHub-Account als Collaborator im Projekt hinterlegt sein):

```
$ git push origin <Branchname>
```

Kollaboratives Arbeiten: Pull request in GitHub

The screenshot shows the GitHub repository interface for 'GerritHeim Material für die Übung'. At the top, there are navigation links for Actions, Projects, Wiki, Security, Insights, and Settings. A yellow notification bar indicates that 'GerritHeim had recent pushes less than a minute ago' with a 'Compare & pull request' button. Below this, the repository details show the 'main' branch with 2 branches and 0 tags. A list of files is displayed, including 'Graf, Fadeeva et al. (Hg.) 2020 - Büch...', 'Literaturliste.csv', 'README.md', and 'Söllner, Mittermaier (Hg.) Mai 2017 - ...'. The 'README.md' file is selected, showing its content: 'Praxislabor digitale Geisteswissenschaften - Einführung in Git'. The README text describes the exercise files and lists three tasks: 'Repositorium clonen', 'Branch erstellen', and 'Pull request erstellen'.

Beitrag von Gerrit Heim #1

This screenshot shows the pull request interface for 'Beitrag von Gerrit Heim #1'. It indicates that 'GerritHeim wants to merge 1 commit into main from GerritHeim'. The pull request details show 0 conversations, 1 commit, 0 checks, and 1 file changed. A comment from GerritHeim, posted 25 seconds ago, asks to take over the contribution change. A green checkmark indicates that 'This branch has no conflicts with the base branch' and that merging can be performed automatically. A 'Merge pull request' button is visible. Below the merge button, there is a text editor for leaving a comment.

Beitrag von Gerrit Heim #1

This screenshot shows the file diff view for 'GerritHeim.md'. It shows 13 lines of text added to the file. The content of the diff is as follows:

```
@@ -0,0 +1,13 @@
1 + DINI-Zertifikat als Sammlung von
2 + Standards und Best Practices
3 + In Deutschland widmet sich das seit 2004 veröffentlichte und inzwischen in der
4 + fünften Auflage erschienene DINI-Zertifikat der Standardisierung von Open-Access-
5 + Repositorien und -Publikationsdiensten.11 Der dem Zertifikat zugrunde liegende Krite-
6 + rienkatalog umfasst etwa 70 Mindestanforderungen sowie zahlreiche Empfehlungen
7 + in insgesamt acht Kriterienbereichen (Müller, Scholze et al., 2016): (1) Sichtbarkeit
8 + des Gesamtangebots, (2) Leitlinien (Policy), (3) Unterstützung für Autor(inn)en und
9 + Herausgeber(innen), (4) Rechtliche Aspekte, (5) Informationssicherheit, (6) Erschlie-
10 + bung und Schnittstellen, (7) Zugriffsstatistik sowie (8) Langzeitverfügbarkeit und
11 + wird durch einen umfangreichen Anhang zur Ausgestaltung der OAI-Schnittstelle
12 + ergänzt. Darin wird unter anderem eine nach fachlichen Gesichtspunkten12 sowie eine
13 + nach formalen Kriterien13 ausgerichtete Set-Struktur definiert. Der Kriterienkatalog
```



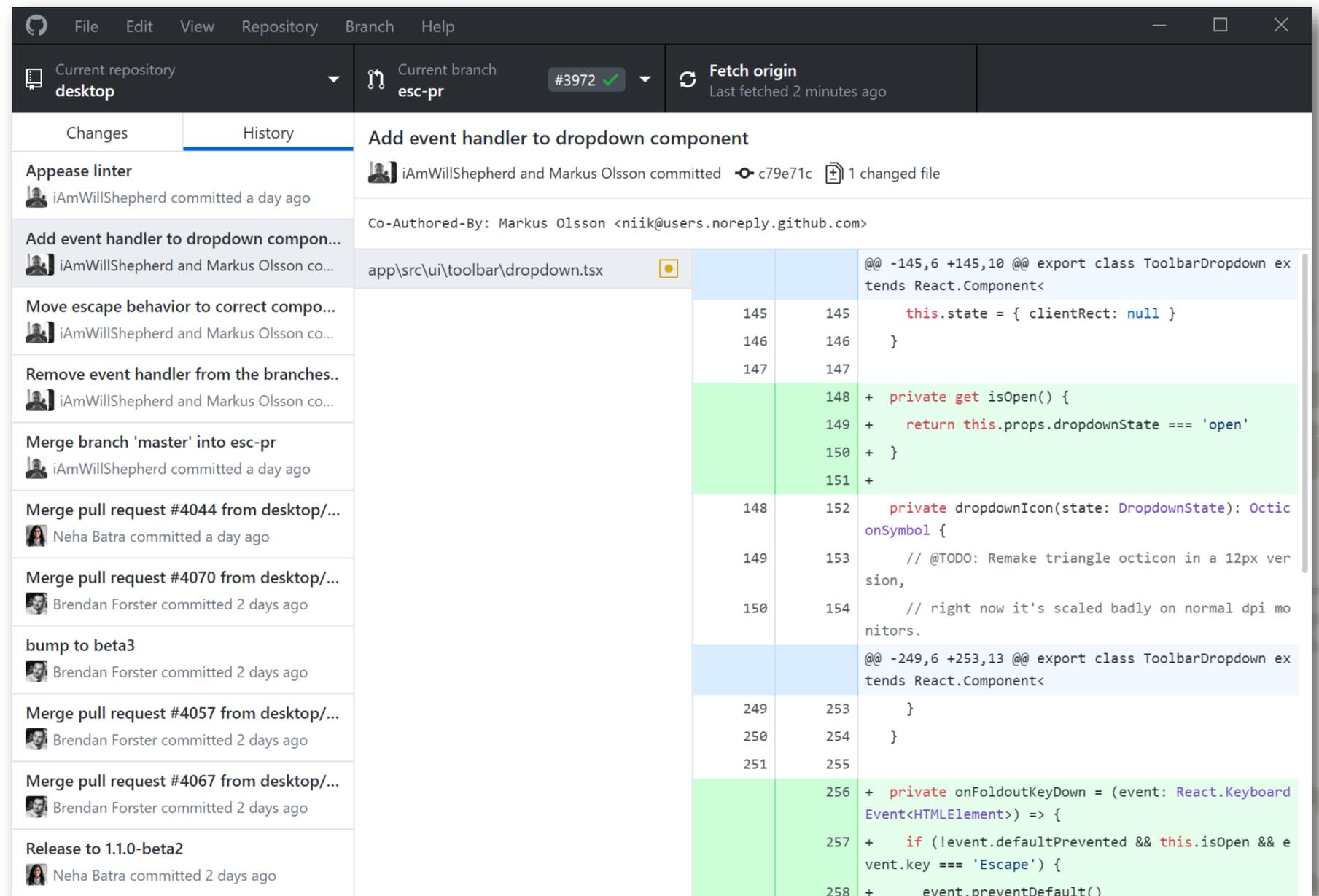
GitHub Desktop

Für Alle, die nach 2 Sitzungen sagen: Nie wieder Konsole!

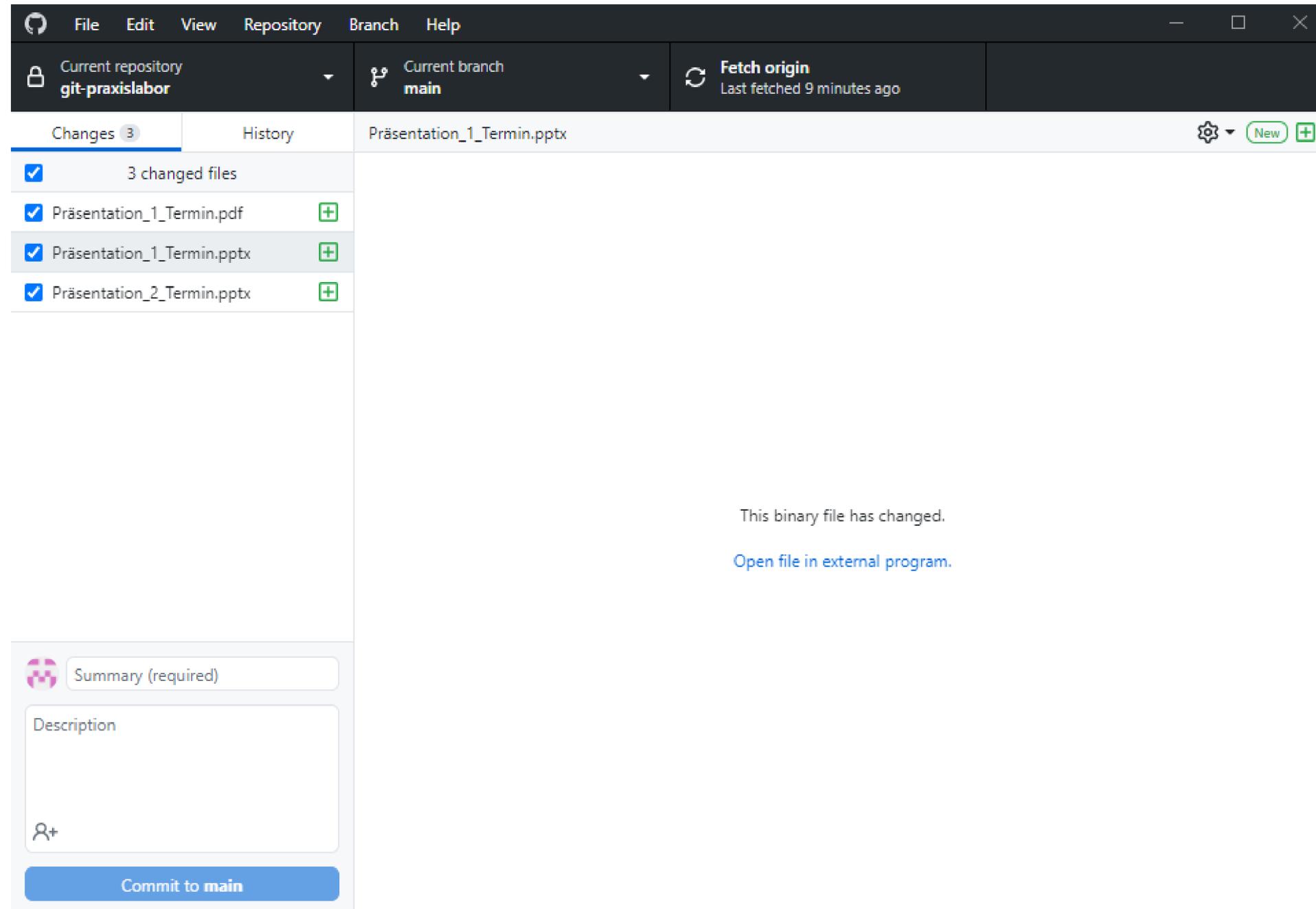
Grafisches Programm für Windows und macOS.

Download:

<https://desktop.github.com/>



GitHub Desktop



The screenshot shows the GitHub Desktop application window. The menu bar includes File, Edit, View, Repository, Branch, and Help. The status bar at the top indicates the current repository is 'git-praxislabor', the current branch is 'main', and the last fetch was 9 minutes ago. The main area is divided into a left sidebar and a main workspace. The sidebar shows a list of 3 changed files: 'Präsentation_1_Termin.pdf', 'Präsentation_1_Termin.pptx', and 'Präsentation_2_Termin.pptx'. The main workspace displays the file 'Präsentation_1_Termin.pptx' and a message: 'This binary file has changed. Open file in external program.' At the bottom left, there is a commit summary section with a 'Summary (required)' field, a 'Description' field, and a 'Commit to main' button.



Weitere grafische Software

- GitGUI (integriert in Git für Windows)
- SmartGit (<https://www.syntevo.com/smartgit/>)
- GitKraken (<https://www.gitkraken.com/>)
- Gitg (<https://wiki.gnome.org/Apps/Gitg/>)
- TortoiseGit (<https://tortoisegit.org/>)
- SourceTree (<https://www.sourcetreeapp.com/>)

usw. usf. → <https://git-scm.com/downloads/guis>





<https://xkcd.com/1597/>

Vielen Dank für die Aufmerksamkeit!

Evaluation: <https://survey.lamapoll.de/Git>



Cheat Sheet Basis

`git init <Pfad>`

z.B. „`git init .`“ für alle geänderten Dateien

`git commit -m „Beschreibung der Änderung“`

Erzeug ein Commit

`git push`

Schiebt die lokalen Änderungen (Commits) auf den Server

`git pull`

Lädt die Änderungen vom Server herunter



Cheat Sheet Basis

`git branch`

Listet alle Branches

`git checkout -b <name>`

Erzeugt einen neuen Branch und wechselt in diesen Branch

