

# Git und GitHub

Angewandte Generative KI und Prompt Engineering  
im Forschungsprojektmanagement  
(Projektmanagement)

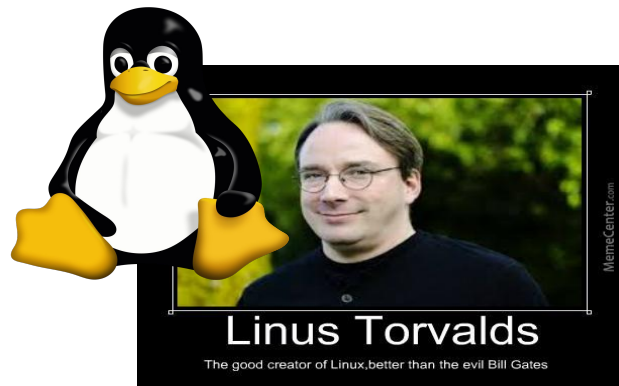
Dr. Christopher Pollin

<https://chpollin.github.io> | [christopher.pollin@dhcraft.org](mailto:christopher.pollin@dhcraft.org)

Digital Humanities Craft OG

[www.dhcraft.org](http://www.dhcraft.org)

Slides were generated AI-assisted.  
Images are partly AI-generated.



*"I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'Git'."*

– Linus Torvalds, <https://de.wikipedia.org/wiki/Git>  
[Git ~ Blödmann]



**Git** ist ein freies, verteiltes Versionsverwaltungssystem für Dateien.



# GitHub



**Datenkrake  
(Microsoft)**



# Lernziele

Die Grundlagen von Git verstehen.

GitHub Desktop kennenlernen und grundlegende Workflows anwenden.

Projektmanagement-Funktionen in GitHub nutzen (Issues, Labels, etc.) und Best Practices erlernen.

Möglichkeiten der Dokumentation in GitHub nutzen und Best Practices kennenlernen.

Ein Verständnis dafür entwickeln, wie Git und GitHub den Projektmanagementprozess in verschiedenen Phasen unterstützen können.

*[Schwerpunkt nicht auf Programmierer\*innen, sondern DH Projekte]*

# Git und seine Grundprinzipien

- **Open HPI: Let's Git - Versionsverwaltung und OpenSource**, <https://open.hpi.de/courses/git2020>  
*[Sehr guter und niederschwelliger Onlinekurs]*
- Blischak, J. D., Davenport, E. R., & Wilson, G. (2016). **A quick introduction to version control with Git and GitHub**. PLoS computational biology, 12(1), e1004668.  
*[Paper für Übersicht]*
- **Git and GitHub Introduction**, [https://www.w3schools.com/git/git\\_intro.asp?remote=github](https://www.w3schools.com/git/git_intro.asp?remote=github)  
*[Tutorial]*
- **Git and GitHub Tutorial For Beginners | Full Course [2021] [NEW]**,  
[https://www.youtube.com/watch?v=3fUbBnN\\_H2c&ab\\_channel=Amigoscode](https://www.youtube.com/watch?v=3fUbBnN_H2c&ab_channel=Amigoscode)  
*[Youtube-Einführung, aber für Programmierer\*innen]*



# Begriffe

## Was ist eine Datei?

z.B. Text (.txt), Bild (.jpg), Musik (.mp3), TEI XML (.xml), JavaScript (.js) ...



## Was ist eine Version?

Eine Version beschreibt den Inhalt einer Datei zu einem bestimmten Zeitpunkt.  
Die Änderung einer Datei erzeugt eine neue Version.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<memo>
```

```
<body>
```

```
<p>Yoda says,
```

```
  "For your support thank I must!"
```

```
</p>
```

```
</body>
```

```
</memo>
```



Version 1

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<memo>
```

```
<body>
```

```
<p>Yoda says,
```

```
<q>For your support thank I must!<q>
```

```
</p>
```

```
</body>
```

```
</memo>
```



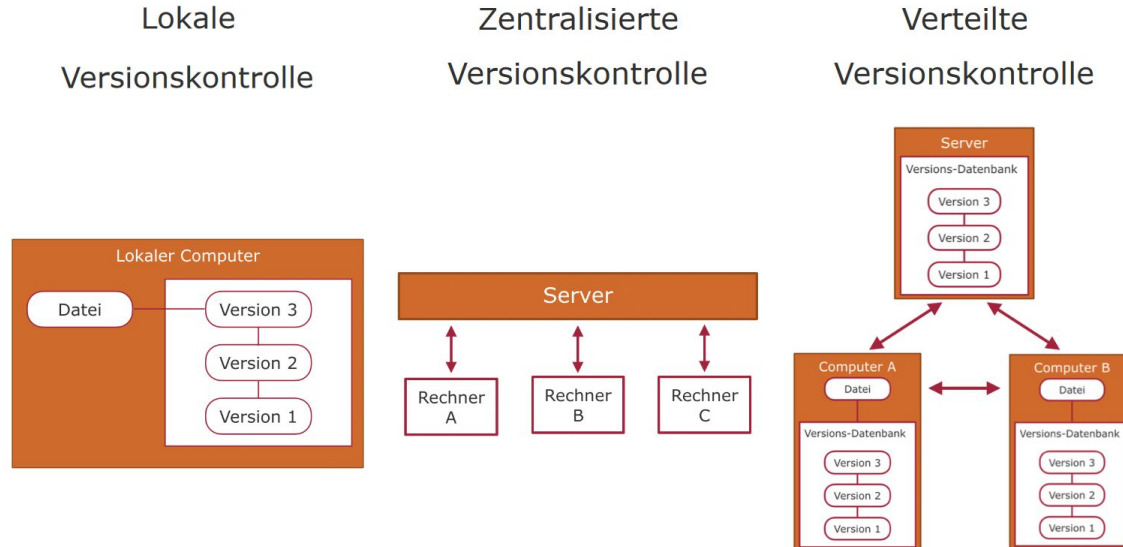
Version 2

# Begriffe

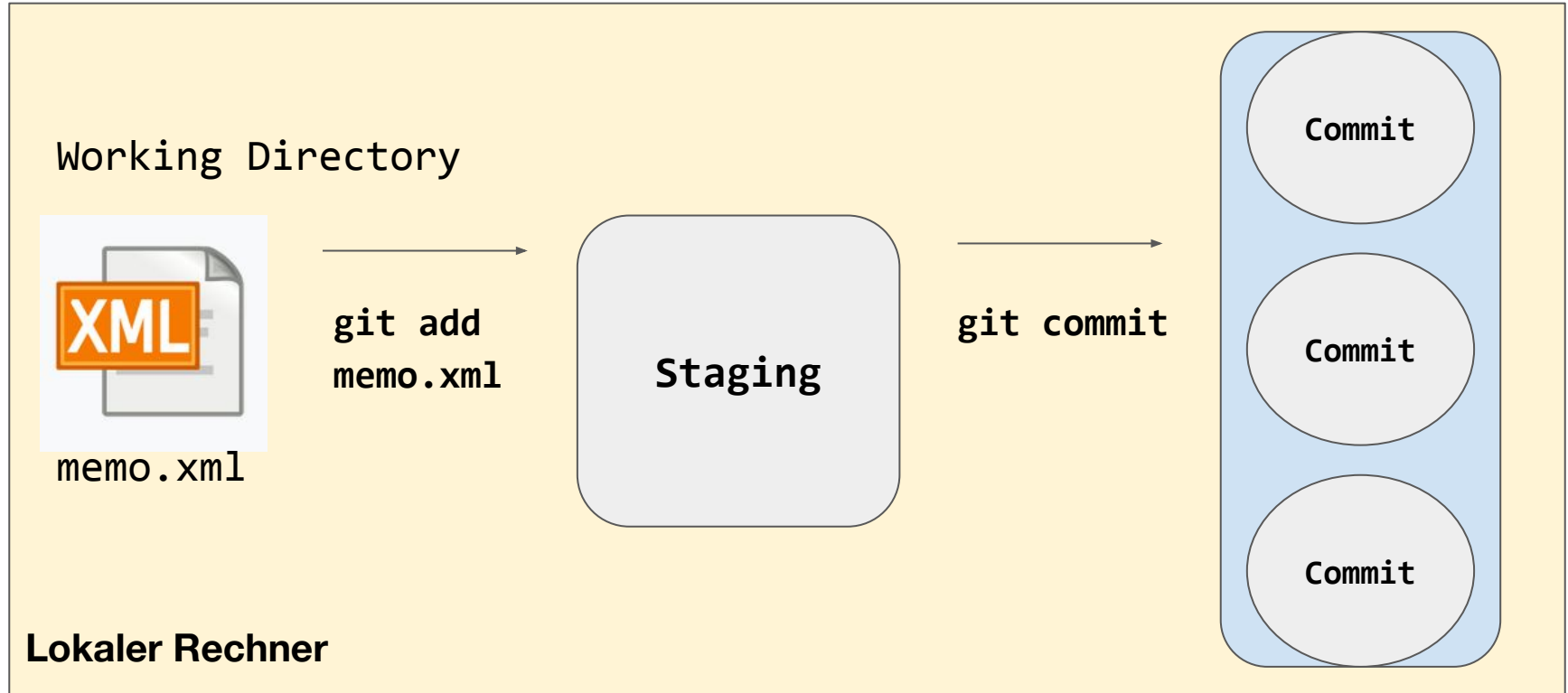
## Was ist Versionskontrolle?

Ermöglicht die Verwaltung verschiedener Versionen von Dateien.

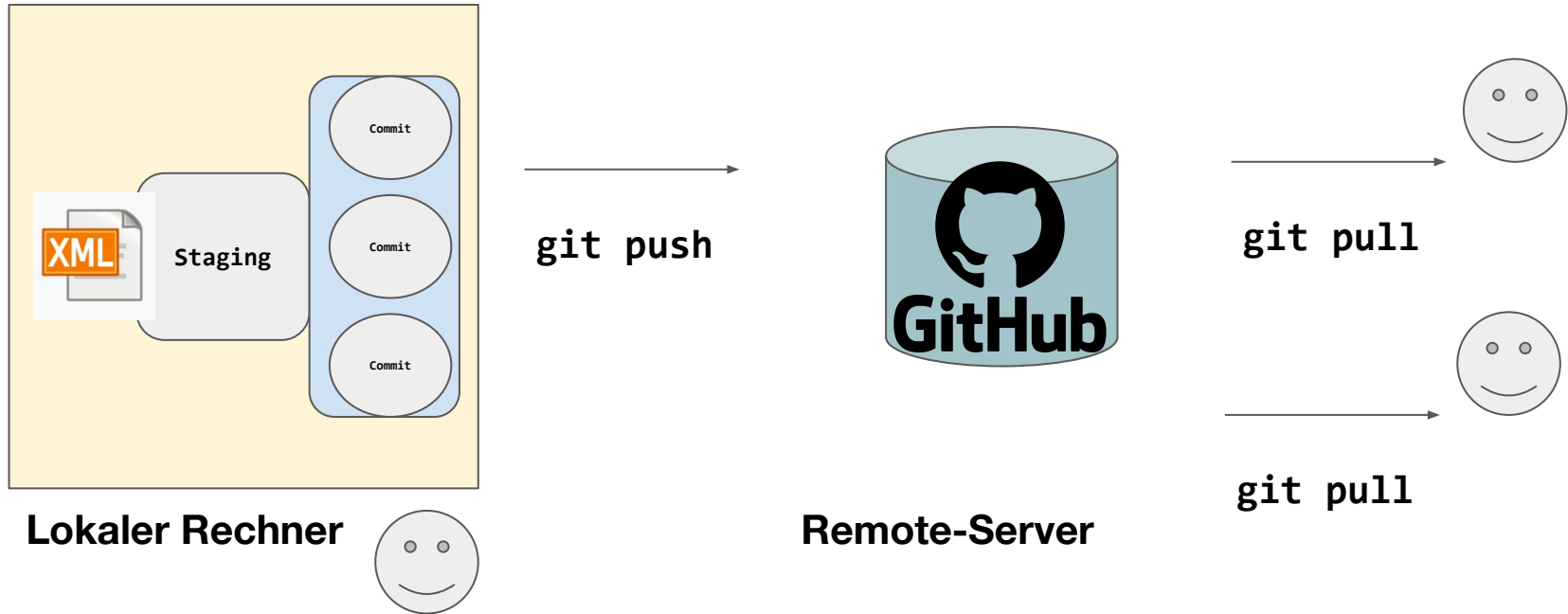
Erlaubt einen “Blick zurück” durch die Versionen von Dateien.



# Git ist ein freies, verteiltes Versionsverwaltungssystem für Dateien



# Lokaler Rechner und Remote-Server





# Git Befehle

## **\$ git add**

Datei (in ihrem gegenwärtigen Zustand)  
zum nächsten Commit hinzufügen (Staging)

## **\$ git commit**

Alles was in Staging ist, kommt als Commit in  
das Repository (Commit History)

## **\$ git push**

Wird verwendet, um Inhalte aus einem lokalen  
Repository in ein Remote-Repository  
hochzuladen.

## **\$ git clone**

Über eine URL ein neues Repository kopieren.

## **\$ git pull**

Der pull-Befehl wird verwendet, um Inhalte aus  
einem Remote-Repository herunterzuladen  
und unverzüglich das lokale Repository zu  
aktualisieren, damit die Inhalte  
übereinstimmen.

## **\$ git fetch**

Damit werden Commits, Dateien und Verweise  
aus einem Remote-Repository in das lokale  
Repository heruntergeladen.

## **\$ git merge**

Die heruntergeladenen Dateien etc. werden mit  
den lokalen zusammengeführt.

# Grundprinzipien von Git

1. Snapshots statt Unterschiede
2. Fast jede Operation ist lokal
3. Git stellt Integrität sicher
4. Git fügt im Regelfall nur Daten hinzu
5. Die drei Zustände (modified, staged, committed)

# Arbeiten mit GitHub

- **GitHub Desktop Quick Intro For Windows**, <https://www.youtube.com/watch?v=77W2JSL7-r8>  
*[Kompaktes Einführungsvideo auf Youtube für GitHub Desktop]*
- **An Intro to Git and GitHub for Beginners (Tutorial)**,  
<https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>  
*[Command line only]*
- **Git for Professionals Tutorial - Tools & Concepts for Mastering Version Control with Git**,  
[https://www.youtube.com/watch?v=Uszj\\_k0DGsg](https://www.youtube.com/watch?v=Uszj_k0DGsg)

# Command Line vs. GUI Clients

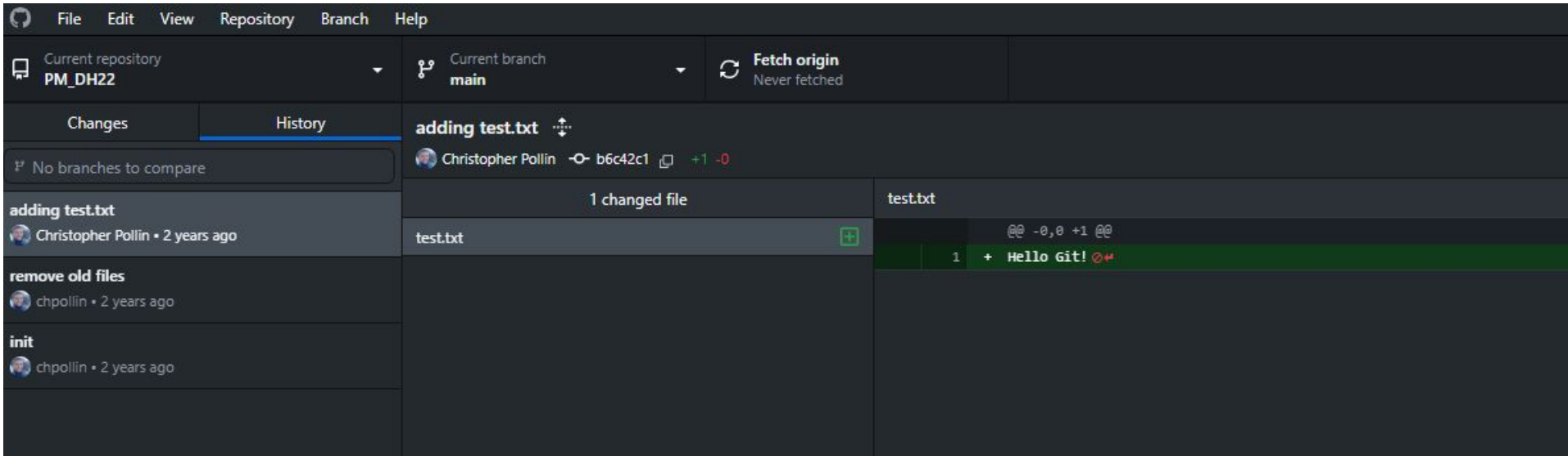
```
Already up to date.  
chpollin@21-NF326:/mnt/c/Users/pollin/Documents/GitHub/SZD$ git --version  
git version 2.25.1  
chpollin@21-NF326:/mnt/c/Users/pollin/Documents/GitHub/SZD$ git pull  
Already up to date.  
chpollin@21-NF326:/mnt/c/Users/pollin/Documents/GitHub/SZD$ |
```

<https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html>

- **GitHub Desktop**  
**Platforms:** Mac, Windows  
**Price:** Free
- **Git Extensions**  
**Platforms:** Linux, Mac, Windows  
**Price:** Free
- **TortoiseGit**  
**Platforms:** Windows  
**Price:** Free
- **GitKraken**  
**Platforms:** Linux, Mac, Windows  
**Price:** Free / \$29 / \$49

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

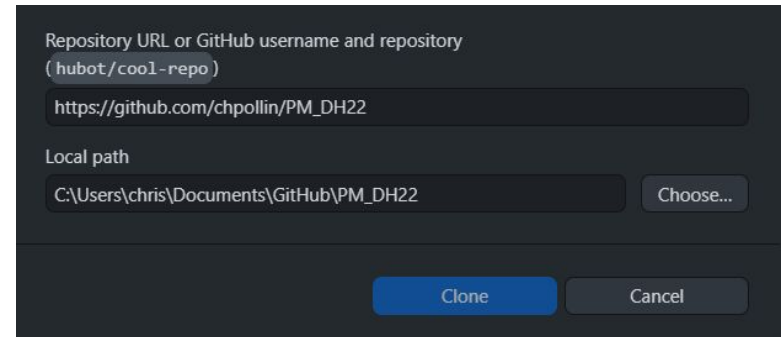
# GitHub Desktop



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

# Gemeinsames Hands-On: “mein memo.xml”

- Erzeugen Sie einen neuen Ordner mit dem Namen GitHub.  
(Z.B.: C:\Users\pollin\Documents\GitHub)
- Legen Sie innerhalb des Ordners GitHub einen Ordner “PM\_DH22” an.
- Öffnen Sie GitHub Desktop.
- Clonen Sie das Repository [https://github.com/chpollin/PM\\_DH22](https://github.com/chpollin/PM_DH22)
  - das Repository heißt “PM\_DH22”.
  - das entspricht dem Befehl **\$ git clone**
- In unserem neuen Ordner befinden sich jetzt alle Daten - lokal bei uns - aus dem Repository.



## Gemeinsames Hands-On: “mein memo.xml”

- Erstellen Sie eine neue XML bzw. verändern Sie helloworld\_template.xml, benennen Sie es um und speichern Sie es - Datei mit folgender Struktur und Ihrem Vornamen.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<memo>
```



```
  <body>
```

```
    <p>Hello Git from Christopher</p>
```

```
  </body>
```

```
</memo>
```

- Legen Sie die Datei mit dem Dateinamen helloworld\_VORNAME.xml in Ihrem lokalen Ordner PM\_DH22\xml ab.

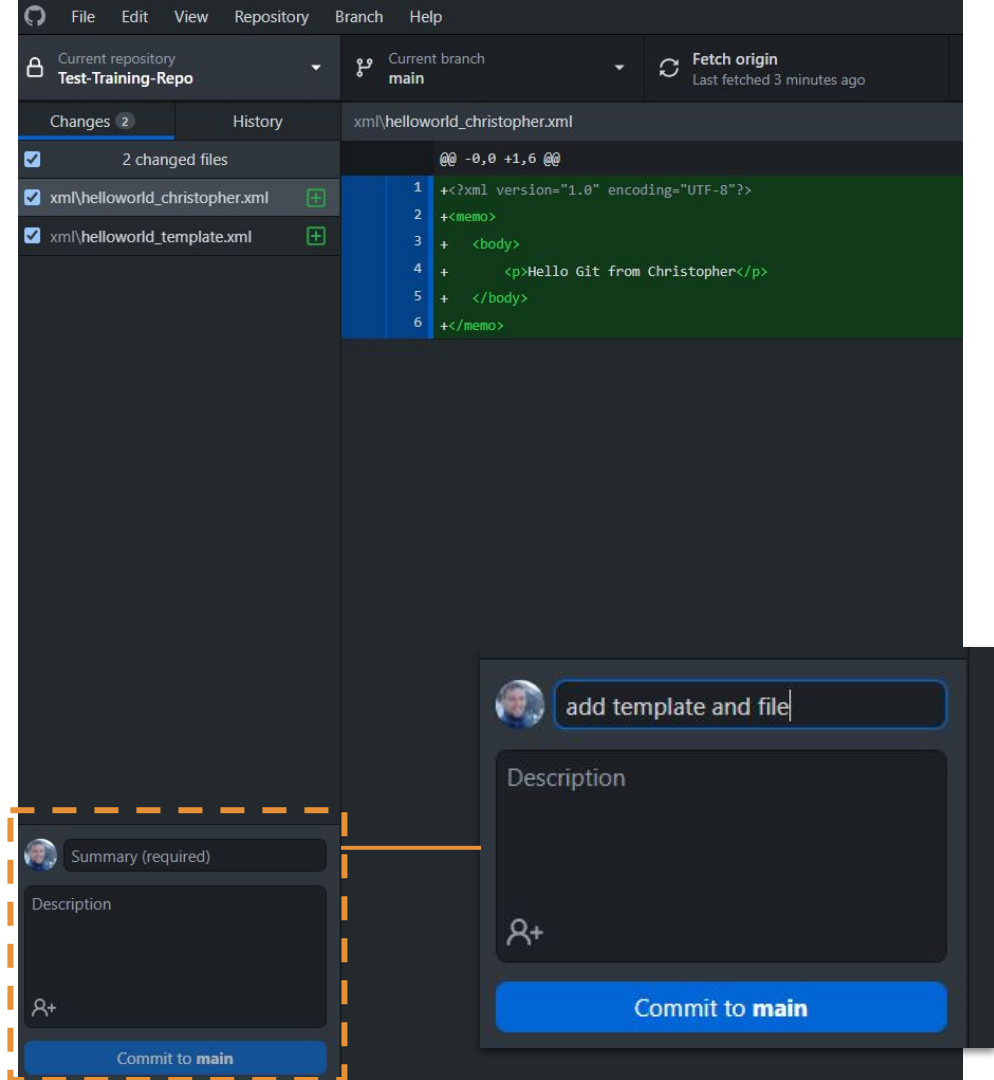
Name	Änderungsdatum	Typ	Größe
 helloworld_christopher	04.07.2022 13:38	XML Document	1 KB
 helloworld_template	04.07.2022 13:38	XML Document	1 KB

Wenn die neue Datei angelegt wurde, erscheinen folgende Änderungen in GitHub Desktop auf.

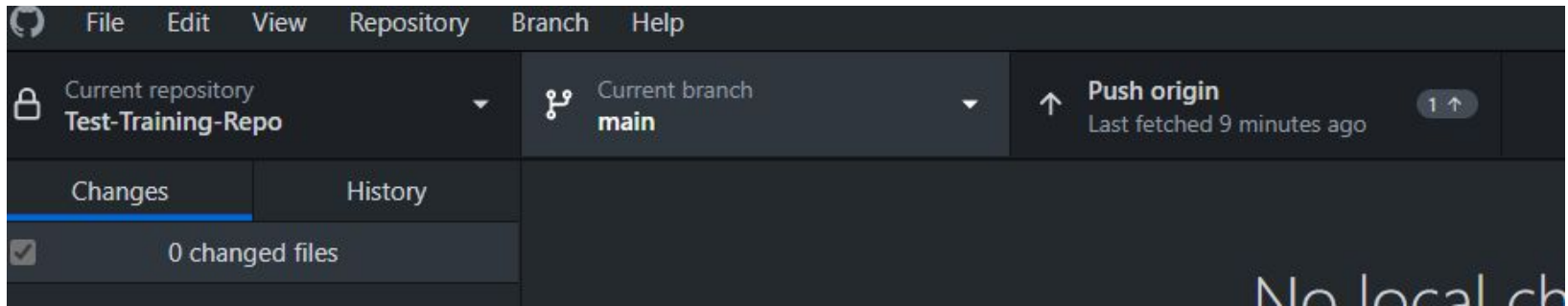
Jeder Commit muss immer eine Nachricht beinhalten. Damit soll ausgedrückt werden, was Sie gerade bearbeitet oder hinzugefügt haben.

Die Dateien befinden sich in der Staging Area.

Mit dem Button “Commit to main” wird git commit aktiviert und der Commit kommt in die Commit History.

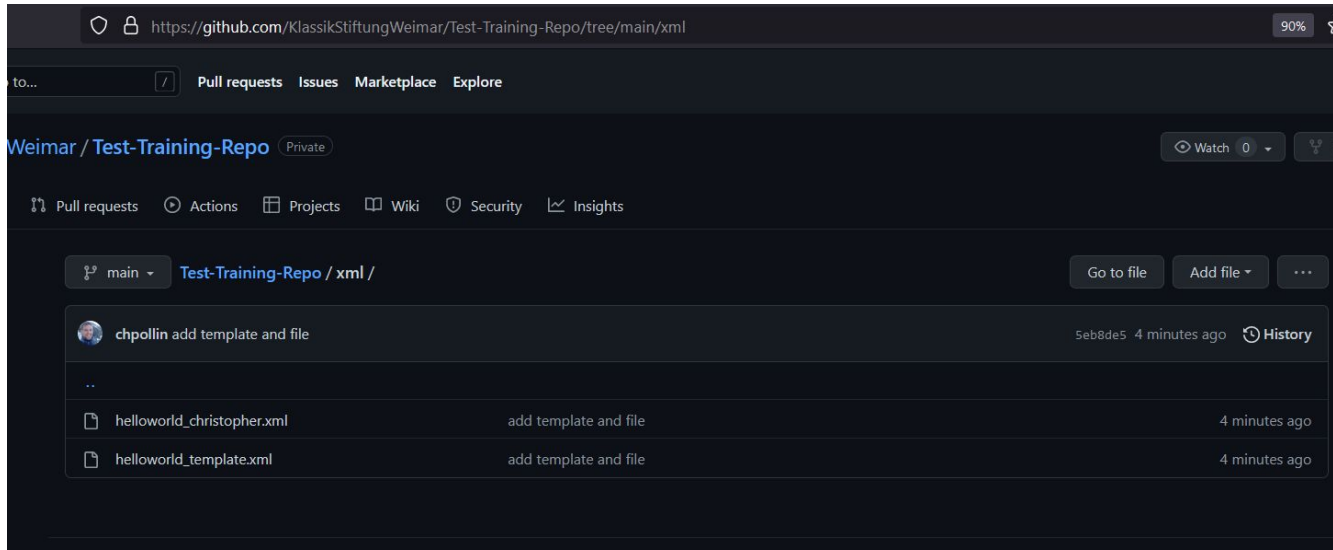




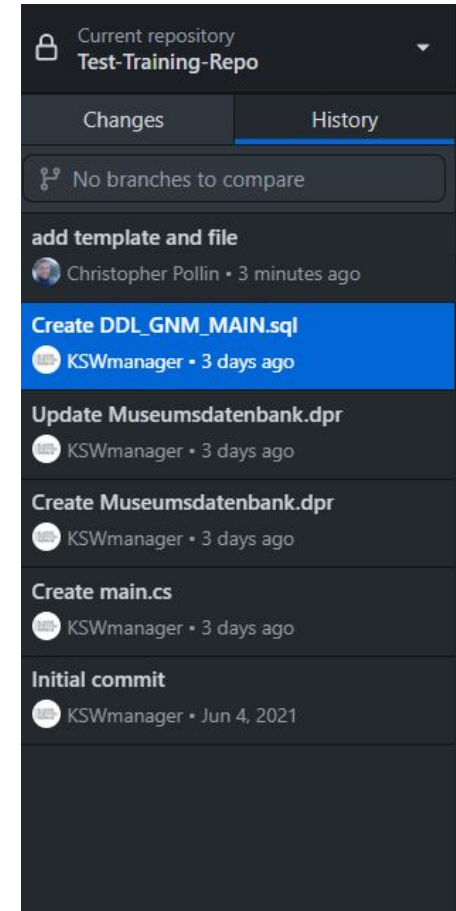


Jetzt muss das Ganze noch in das Repository (Test-Training-Repo) gepusht werden - `git push` aktiv.

Unsere Dateien sind jetzt auf dem Server.

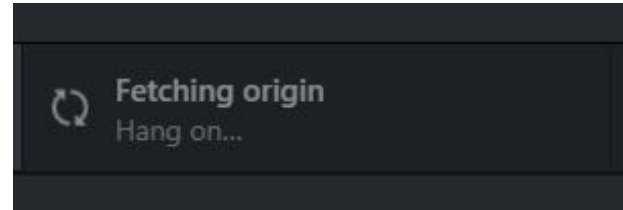
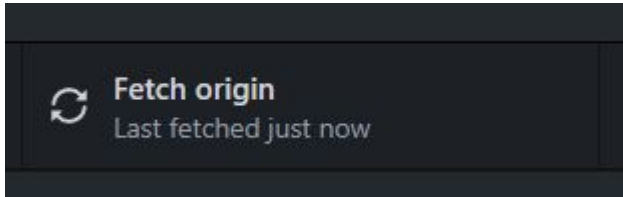


In der History in GitHub Desktop bzw. auf der Webseite  
[https://github.com/chpollin/PM\\_DH22/tree/main/xml](https://github.com/chpollin/PM_DH22/tree/main/xml)  
können wir die Commits einsehen.



**\$ git fetch**

Gleichen Sie ab: Gibt es was Neues bei uns lokal? Gibt es etwas Neues im Repository am Server?

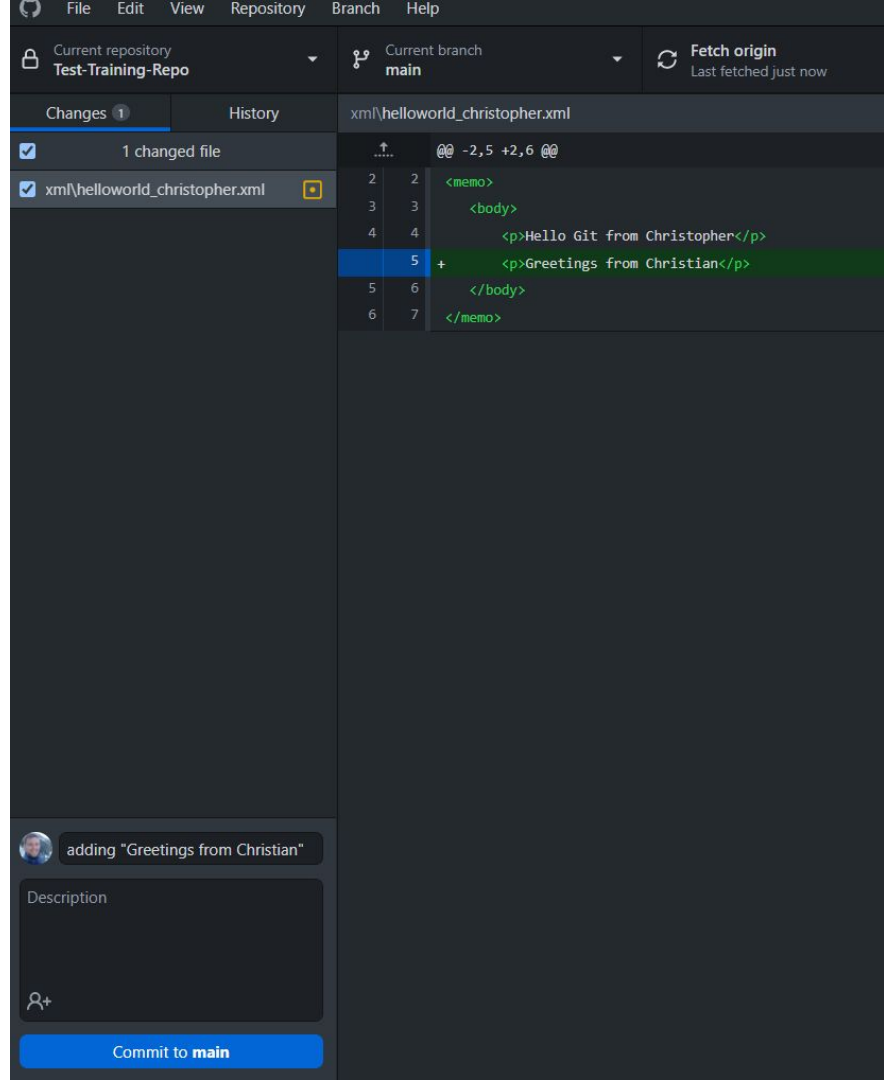


Bearbeiten Sie nun das XML-Dokument von Ihrer/Ihrem alphabetisch nächsten Kollegin/Kollegen. Fügen Sie eine Kleinigkeit hinzu, wie z.B. ein zweites `<p>` mit dem Text “Greetings from Christian”.


```
<?xml version="1.0" encoding="UTF-8"?>
<memo>
  <body>
    <p>Hello Git from Christopher</p>
    <p>Greetings from Christian</p>
  </body>
</memo>
```

Es könnte jetzt zu einem Konflikt gekommen sein, weil zwei Personen gleichzeitig dieselbe Datei bearbeitet haben.

“*Git can handle most merges on its own with automatic merging features. **A conflict arises when two separate branches have made edits to the same line in a file, or when a file has been deleted in one branch but edited in the other.** Conflicts will most likely happen when working in a team environment.*”




← → ↻ <https://github.com/KlassikStiftungWeimar/Test-Training-Repo> 90% ☆

 Search or jump to... [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


[KlassikStiftungWeimar / Test-Training-Repo](#) Private [Watch](#) 0 [Fork](#) 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

[main](#) [1 branch](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

 **chpollin** adding "Greetings from Christian" 27c7f65 22 minutes ago ⌚ 7 commits

xml	adding "Greetings from Christian"	22 minutes ago
DDL_GNM_MAIN.sql	Create DDL_GNM_MAIN.sql	3 days ago
Museumsdatenbank.dpr	Update Museumsdatenbank.dpr	3 days ago
README.md	Initial commit	12 months ago
main.cs	Create main.cs	3 days ago

README.md 

# Test

Test

**About**

**Test**

[Readme](#)

☆ 0 stars

👁 0 watching

🍴 0 forks

**Releases**

No releases published  
[Create a new release](#)

**Packages**

No packages published  
[Publish your first package](#)

**Contributors** 2

[https://github.com/chpollin/PM\\_DH22](https://github.com/chpollin/PM_DH22)

# “The Perfect Commit”

- Jeder Commit gehört zu einem Thema. Themen sollten nicht in den gleichen Commits gemischt werden.
  - Commit: Annotation der Personen im XML/TEI
  - Überarbeitung des msDesc im teiHeader
  - Das XSLT zur Normalisierung von X→ nur die Dateien ankreuzen, die zum Thema passen.  
→ gemeinsame Regeln definieren, was ein Commit ist.
- Commit-Message
  - Kleine Zusammenfassung
  - “Kurz und knackig” & effektiv
  - Ungewöhnliches und Wichtiges explizieren→ gemeinsame Regeln definieren, wie eine Commit-Message aufgebaut sein muss.

# GitHub als Software Engineering Tool

- **GitHub Projects Demo: Automation, Kanban, Scrum, Issues, Pull Request, Milestones, Issues, Tasks,** <https://www.youtube.com/watch?v=YxKhb3fxtsU>
- **GitHub Issues,** <https://docs.github.com/en/enterprise-server@3.5/issues>
- **Labels and Milestones,** <https://docs.github.com/en/enterprise-server@3.5/issues/using-labels-and-milestones-to-track-work/managing-labels>
- **Managing Project Boards,** <https://docs.github.com/en/enterprise-server@3.5/issues/organizing-your-work-with-project-boards/managing-project-boards/about-project-boards>

# Issues

The screenshot shows a GitHub repository interface. At the top, navigation links include 'Issues' (26), 'Pull requests', 'Actions', 'Projects' (2), 'Wiki', 'Security', and 'Insights'. The main heading is 'Beispiel Issue #3'. Below it, a green 'Open' button and a badge '1 of 3 tasks' are visible, along with the text 'chpollin opened this issue on Jun 14 · 1 comment'.

A comment by 'chpollin' from June 14 is shown. The comment header includes the user's profile picture and the text 'chpollin commented on Jun 14 · edited'. The comment body contains:

- A title 'Beispiel Issue'.
- A paragraph: 'Hier ist die Beschreibung der Task, die bearbeitet werden muss.'
- A paragraph: 'Diese könnte man in weitere, kleinere Aufgaben unterteilen und schön mit einer task list dokumentieren'.
- A task list:
  - ☒ Christian macht X im teilHader
  - ☐ @chpollin macht Y im body
  - ☐ zusammenführen
- A paragraph: 'Auch Screenshots können helfen Dinge zu visualisieren. Unter Windows einfach mit dem Snipping Tool einen Screenshot erstellen und mit strg-v hier direkt einfügen.'
- A code block containing XML-like markup:

```
</msIdentifier>
</msDesc>
<listBibl>
  <bibl xml:id="bibl.1"><title>Lorem Ipsum 1</title></bibl>
  <bibl xml:id="bibl.2"><title>Lorem Ipsum 2</title></bibl>
  <bibl xml:id="bibl.3"><title>Lorem Ipsum 3</title></bibl>
</listBibl>
</sourceDesc>
```

<https://github.com/KlassikStiftungWeimar/Test-Training-Repo/issues/3>



# Milestones

The screenshot shows the GitHub interface for a milestone titled "TEI Model". At the top, navigation links include Issues (6), Pull requests, Actions, Projects, Wiki, Security, and Insights. Below these, tabs for "Labels" and "Milestones" are visible, with "Milestones" being the active tab. The milestone title "TEI Model" is displayed in large text, followed by a green progress bar. Below the bar, it states "Due by July 11, 2022" and "33% complete". The description of the milestone is "Planung und Konzeption des TEI Models für X.". A summary bar indicates "2 Open" and "1 Closed" issues. Two issues are listed: "TEI Model: Reading" (opened 4 minutes ago by chpollin, 3 of 4 tasks) and "TEI Model: Prototype" (marked as priority, opened 3 minutes ago by chpollin). A tip at the bottom suggests using keyboard shortcuts to move items.

Issues 6 Pull requests Actions Projects Wiki Security Insights

Labels Milestones

## TEI Model

Due by July 11, 2022 33% complete

Planung und Konzeption des TEI Models für X.

2 Open 1 Closed

- TEI Model: Reading  
#6 opened 4 minutes ago by chpollin 3 of 4 tasks
- TEI Model: Prototype **priority**  
#7 opened 3 minutes ago by chpollin

Tip! You can use `shift+j` or `shift+k` to move items with your keyboard.

[https://github.com/chpollin/PM\\_DH22/milestone/1](https://github.com/chpollin/PM_DH22/milestone/1)

## Beispiele

Consulting Schönberg Edition

<https://github.com/chpollin/Schoenberg/issues>

Interne Arbeit in DHCraft

<https://github.com/DigitalHumanitiesCraft/Projects/issues>

Stefan Zweig digital

<https://github.com/chpollin/SZD/issues>

# Projects

The screenshot shows the GitHub Projects interface for the repository 'KlassikStiftungWeimar / Test-Training-Repo'. The repository is marked as 'Private'. The navigation bar includes links for Code, Issues (6), Pull requests, Actions, Projects (1), Wiki, Security, and Insights. The Projects section is active, displaying a Kanban board with three columns: 'todo' (2 items), 'in progress' (1 item), and 'done' (0 items). The 'todo' column contains two cards: 'add new <p> to helloworld\_christopher.xml' (opened by chpollin) and 'TEI Model: Reading' (3 of 4 tasks, opened by chpollin). The 'in progress' column contains one card: 'Beispiel Issue' (1 of 3 tasks, opened by chpollin). The 'done' column is empty. On the right, a sidebar shows search results for the filter 'is:open', listing three issues: 'TEI Model: Prototype' (priority, opened by chpollin), 'teiBody: normalize <persName>' (opened by chpollin), and 'teiHeader: add msDesc' (opened by chpollin).

KlassikStiftungWeimar / Test-Training-Repo (Private)

Watch 2 Fork 0 Star 0

Code Issues 6 Pull requests Actions Projects 1 Wiki Security Insights

test Updated now

Filter cards

2 todo + ...

- add new <p> to helloworld\_christopher.xml #9 opened by chpollin
- TEI Model: Reading 3 of 4 tasks #6 opened by chpollin

1 in progress + ...

- Beispiel Issue 1 of 3 tasks #3 opened by chpollin

0 done + ...

+ Add cards

is:open

You can use the filters available in issue search.

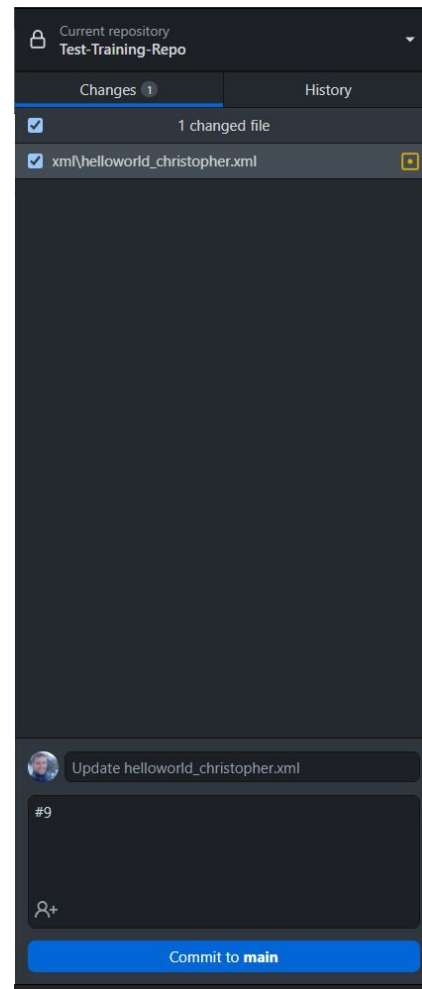
Search results

- TEI Model: Prototype #7 opened by chpollin (priority) TEI Model
- teiBody: normalize <persName> #5 opened by chpollin (teiBody)
- teiHeader: add msDesc #4 opened by chpollin (teiHeader)

<https://github.com/KlassikStiftungWeimar/Test-Training-Repo/projects/1>

# Hands-On: Issue anlegen

- Legen Sie ein Issue an und assignen Sie jemanden, der/die eine Änderung in einem der vielen helloworld\_xxx.xml machen soll.
- Fügen Sie ein Label und einen Milestone hinzu.
- Die zugeordnete Person führt den Issue aus, committet die Änderung und referenziert mit #ISSUE ID auf das zugehörige Issue.
- Pushen Sie die Änderung.
- Fügen Sie ein Comment hinzu und schließen Sie das Issue.



## **“The best Issue”**

Simple, prägnant und beschreibend

Einheitliche Formatierung

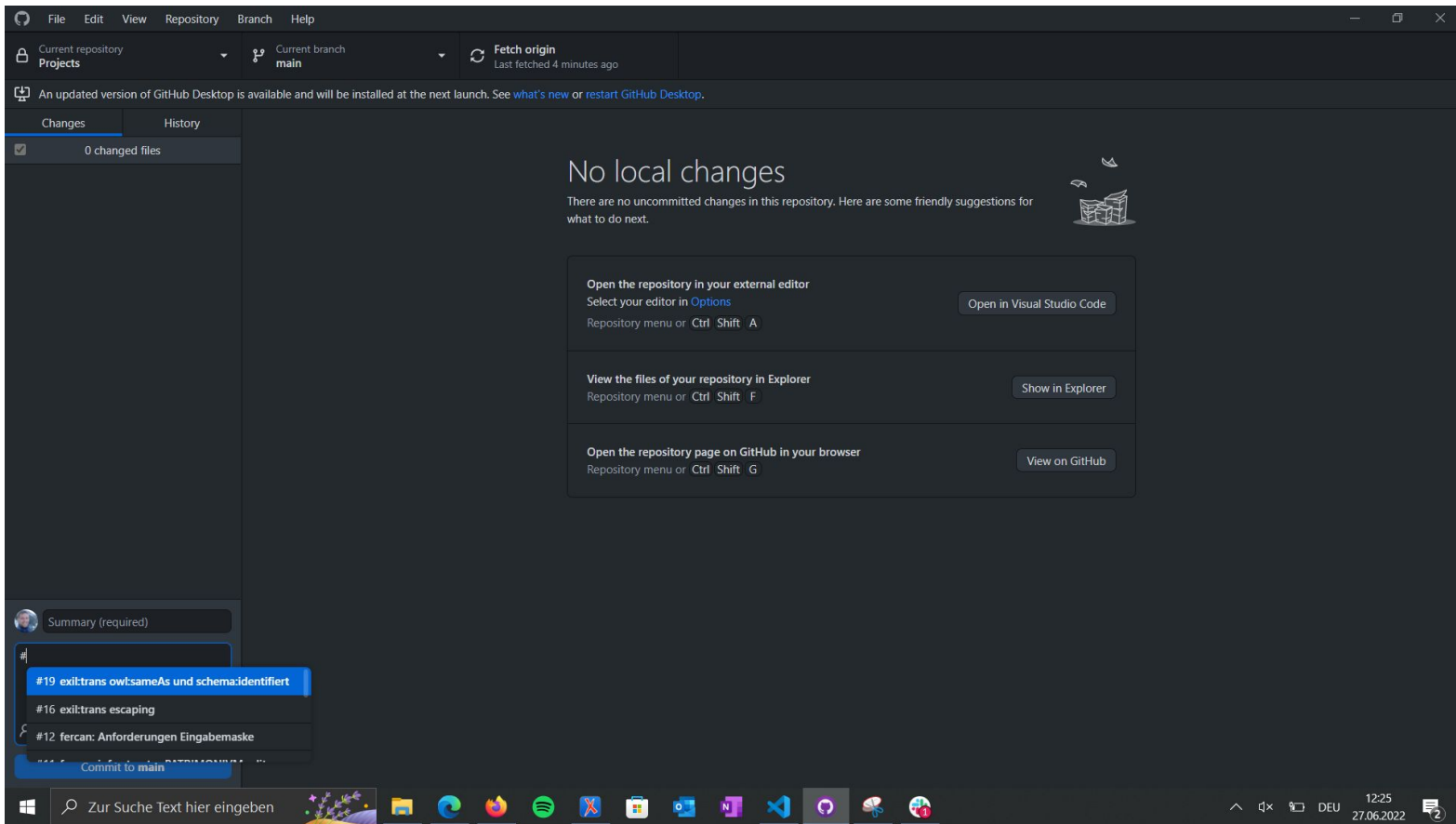
Bearbeitbar / Durchführbar

Relevant

Kontext

Problem oder Idee

Lösung und nächste Schritte



- <https://github.com/DigitalHumanitiesCraft/Projects/commit/5e9e5c05eccc19fa2eca9d45456ac3f89b29e8c5>
- <https://github.com/DigitalHumanitiesCraft/Projects/issues/18>

# GitHub und Dokumentation

- **Introduction to Markdown**, <https://www.geeksforgeeks.org/introduction-to-markdown/>
- **Markdown Cheat Sheet. A quick reference to the Markdown syntax**, <https://www.markdownguide.org/cheat-sheet/>
- **Markdown Tips & Tricks 2022 - Markdown Crash Course**, <https://www.youtube.com/watch?v=ftOBvusMHjQ>
- **Markdown Crash Course**, <https://github.com/codeSTACKr/markdown-crash-course>

# Markdown

cheatsheet.md [C:\Users\pollin\Documents\GitHub\KlassikStiftungWeimar\Test-Training-Repo\documents\cheatsheet.md] - <oXygen/> XML Editor (Academic use only)

File Edit Find Project Options Tools Document Window Help

XPath 2.0 Execute XPath on 'Current File'

```
1 # Heading 1
2
3 Text
4
5 * item
6 * item
7 * item
8
9
10 ```
11 <memo>
12   <from>Christopher</from>
13   <to>Christian</to>
14 </memo>
15 ```
16
17
18 ## Heading 2
19
20 ### Heading 3
```

Heading 1

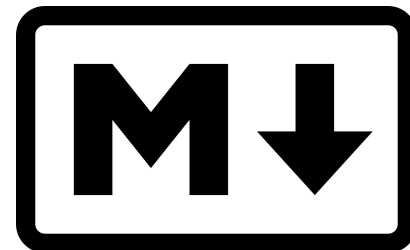
Text

- item
- item
- item

```
<memo>
  <from>Christopher</from>
  <to>Christian</to>
</memo>
```

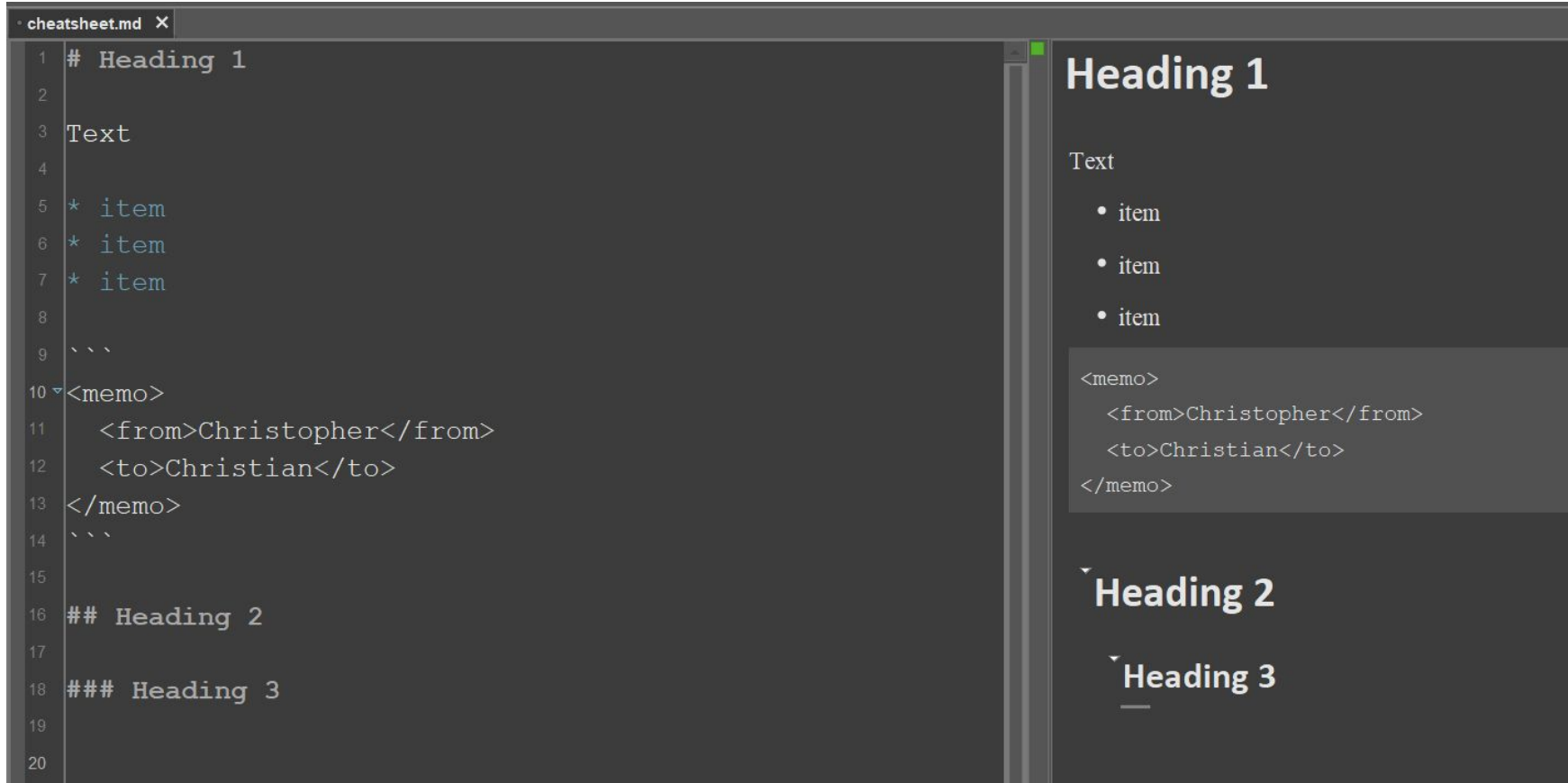
Heading 2

Heading 3





# Überschriften, Text, ungeordnete Liste, Code Block



The image shows a side-by-side comparison of Markdown source code and its rendered HTML output. The left pane is a code editor with a dark theme, showing a file named 'cheatsheet.md'. It contains 20 lines of code. The right pane shows the rendered result of this code, with a dark background and white text. The rendered output includes a large heading, a paragraph of text, a bulleted list, a preformatted code block (a memo), and two more headings of different sizes.

```
1 # Heading 1
2
3 Text
4
5 * item
6 * item
7 * item
8
9 ```
10 <memo>
11   <from>Christopher</from>
12   <to>Christian</to>
13 </memo>
14 ```
15
16 ## Heading 2
17
18 ### Heading 3
19
20
```

## Heading 1

Text

- item
- item
- item

```
<memo>
  <from>Christopher</from>
  <to>Christian</to>
</memo>
```

## Heading 2

### Heading 3

# Absätze, geordnete Liste, Schriftauszeichnung (kursiv & fett)

```
18 ### Heading 3
19
20 Lorem ipsum dolor sit amet, consetetur sadipscing
21 elitr, sed diam nonumy eirmod tempor
22 invidunt ut labore et dolore magna aliquyam
23 erat, sed diam voluptua.
24 At vero eos et accusam et justo
25 duo dolores et ea rebum.
26
27 Erst eine zusätzliche Zeile macht einen neuen Absatz.
28
29 ## Ordered List
30 1. *item 1 is italic*
31 2. **item 2 is bold**
32 3. ***item 3*** is italic and bold
33
34 Aber auch so:
35
36 1. item 1
37 1. item 2
38 3. item 3
```

## Heading 3

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Erst eine zusätzliche Zeile macht einen neuen Absatz

## Ordered List

1. *›item 1 is italic*
2. **›item 2 is bold**
3. ***››item 3 is italic and bold***

Aber auch so:

1. item 1
2. item 2
3. item 3

# Blockzitate, Links, Bilder, Tabellen, Inhaltsverzeichnis

```
39 2. Item 4
40
41 ### Blockquote
42
43 > some text
44 >
45 > some text
46
47 ### Links
48
49 Das ist ein Link: [text you see](https://de.wikipedia.org/wiki/Markdown)
50
51 Das ist ein interner Link: [Spring zu Heading 1](#heading1)
52
53 ![nur ein Beispiel](https://commons.wikimedia.org/wiki/File:Example_de.jpg "Beispielbild")
54
55 ## Extended Syntax
56
57 ### Tabelle
58
59 | Packages | Description | Version |
60 | :--- | :---: | ---: |
61 | React | JavaScript Framework | v18.0 |
62 | Next.js | React Framework | v12.0 |
63
64
65
66
67
68
```

## Blockquote

some text

some text

## Links

Das ist ein Link: [text you see](https://de.wikipedia.org/wiki/Markdown)

Das ist ein interner Link: [Spring zu Heading 1](#)

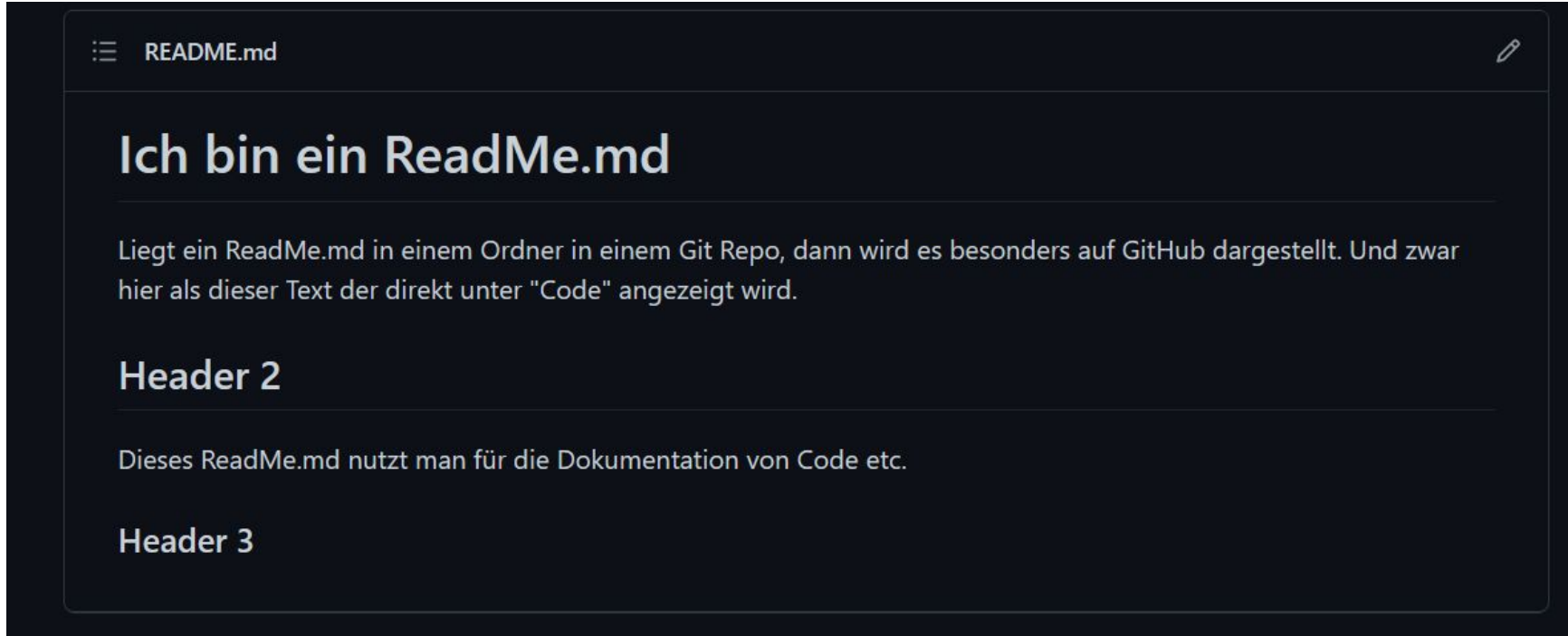
[nur ein Beispiel](#)

## Extended Syntax

### Tabelle

Packages	Description	Version
React	JavaScript Framework	v18.0
Next.js	React Framework	v12.0

# README.md zur Dokumentation



[https://github.com/chpollin/PM\\_DH22](https://github.com/chpollin/PM_DH22)

About READMEs,

<https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>

# Wiki: Beispiel Kodierungsregeln

The screenshot shows a GitHub Wiki page for a project. The top navigation bar includes links for Issues (5), Pull requests, Discussions, Actions, Projects, Wiki (active), Security, Insights, and Settings. The page title is 'Kodierungsregeln' with an 'Edit' button. Below the title, it says 'phkehrer edited this page on Mar 3 · 46 revisions'. The main content is divided into two sections: 'Entitäten (durch Registerfunktion)' and 'Hervorhebungen'. The first section lists six items with their corresponding XML-like tags and a reference to '[ediarum.Base]'. The second section lists 'Fett' and 'Unterstrichen' with their respective tags and a reference. A right sidebar shows a 'Pages' section with a search bar and a list of pages, including 'Home' and 'Kodierungsregeln' (which is highlighted).

Issues 5 Pull requests Discussions Actions Projects Wiki Security Insights Settings

## Kodierungsregeln

phkehrer edited this page on Mar 3 · 46 revisions

Edit

### Entitäten (durch Registerfunktion)

- Datum: `<date when="1900-01-01" calendar="#Gregorian"/>` [ediarum.Base]
- Personenname: `<persName key="person.2">Emil Hertzka</persName>` [ediarum.Base]
- Organisationsname: `<orgName key="org.3">Kolisch-Quartett</orgName>` [ediarum.Base]
- Ortsname: `<placeName key="place.1.1">Wien</placeName>` [ediarum.Base]
- Werktitel: `<bibl sameAs="werke.31">Komm, Gott, Schöpfer, Heiliger Geist</bibl>` [ediarum.Base]
- Sachbegriffe:

### Hervorhebungen

- Fett: `<hi rendition="#b">` [ediarum.Base]
- Unterstrichen: `<hi rendition="#u">` [ediarum.Base]
- `<hi rendition="#u">Bläser-Partituren-Stimmen</hi>` [ediarum.Base]

Pages 2

Find a Page...

Home

Kodierungsregeln

- Entitäten (durch Registerfunktion)
- Hervorhebungen
- Streichungen
- Hinzufügungen
- Lesbarkeit
- Korrektur durch den Herausgeber
- Normalisierte Schreibung

<https://github.com/chpollin/Schoenberg/wiki/Kodierungsregeln>

About wikis, <https://docs.github.com/en/communities/documenting-your-project-with-wikis/about-wikis>

# Hands-On: Markdown

Erstellen Sie eine neue Wiki Page (mit ihrem Namen). Probieren Sie Markdown aus → Überschrift, Liste und eine Code Snippet (memo.xml).

The screenshot shows the GitHub interface for the repository 'KlassikStiftungWeimar / Test-Training-Repo'. The repository is marked as 'Private'. The navigation bar includes links for Code, Issues (6), Pull requests, Actions, Projects (1), Wiki (selected), Security, and Insights. On the right, there are buttons for Watch (2), Fork (0), and Star (0). The main content area is titled 'Home' and shows a message: 'Christopher Pollin edited this page now · 1 revision'. Below this, it says 'Welcome to the Test-Training-Repo wiki!'. There is a dashed box with a '+ Add a custom footer' button. On the right side, there is a 'Pages' section with a count of 1, a search bar labeled 'Find a Page...', and a link to 'Home'. At the bottom right, there is another dashed box with a '+ Add a custom sidebar' button.

KlassikStiftungWeimar / Test-Training-Repo Private

Watch 2 Fork 0 Star 0

<> Code Issues 6 Pull requests Actions Projects 1 Wiki Security Insights

## Home

Christopher Pollin edited this page now · 1 revision

Welcome to the Test-Training-Repo wiki!

+ Add a custom footer

Pages 1

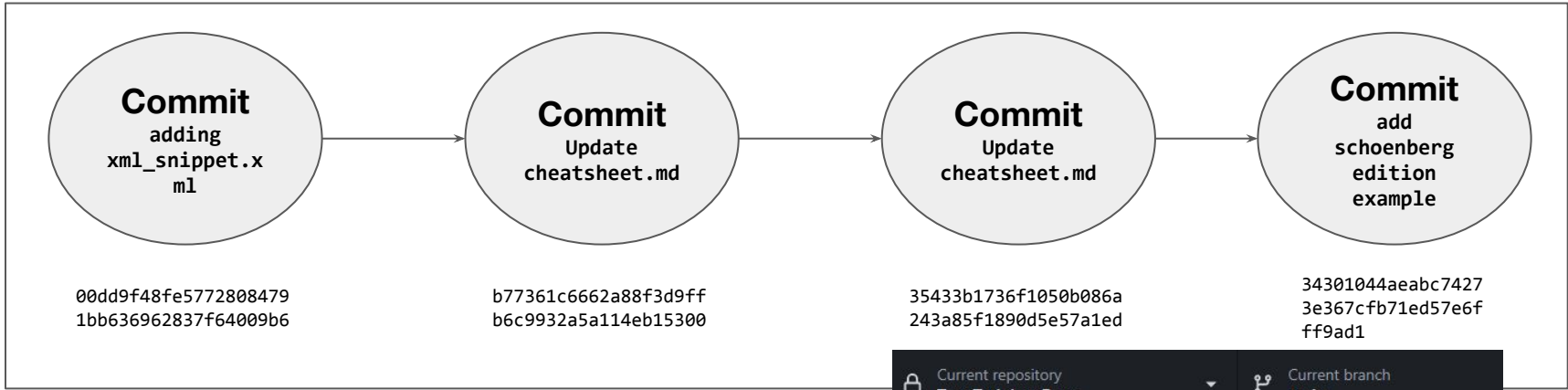
Find a Page...

Home

+ Add a custom sidebar

# **Git Branches**

# Branches



main

Current repository  
Test-Training-Repo

Current branch  
main

Changes

History

No branches to compare

add schoenberg edition example

Pollin • 8 hours ago

Update cheatsheet.md

Pollin • 11 hours ago

Update cheatsheet.md

Pollin • 11 hours ago

adding xml\_snippet.jpg

Pollin • 11 hours ago

add schoenberg edition exam...

Pollin 3430104 30 cha

digital\_edition\_schoenberg\_...(\Org

digital\_edition\_schoenberg\_exam

digital\_edition\_schoenberg\_exam

digital\_edition\_schoenberg\_exa...(\

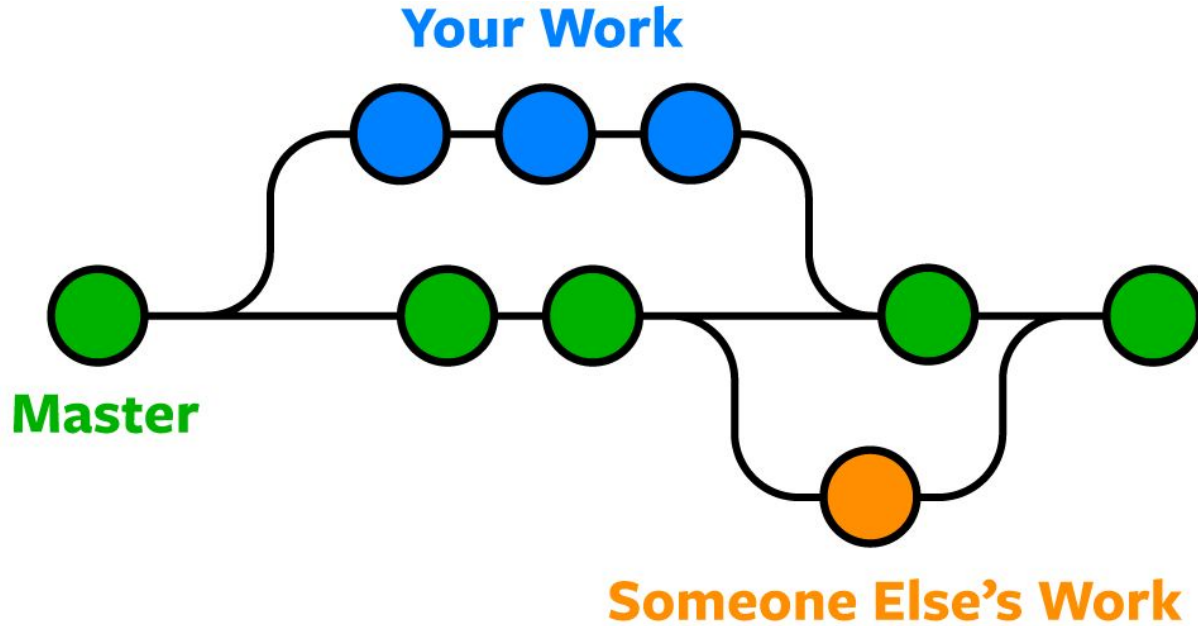
digital\_edition\_schoenberg\_exam

digital\_edition\_schoenberg\_exam

digital\_edition\_schoenberg\_exam

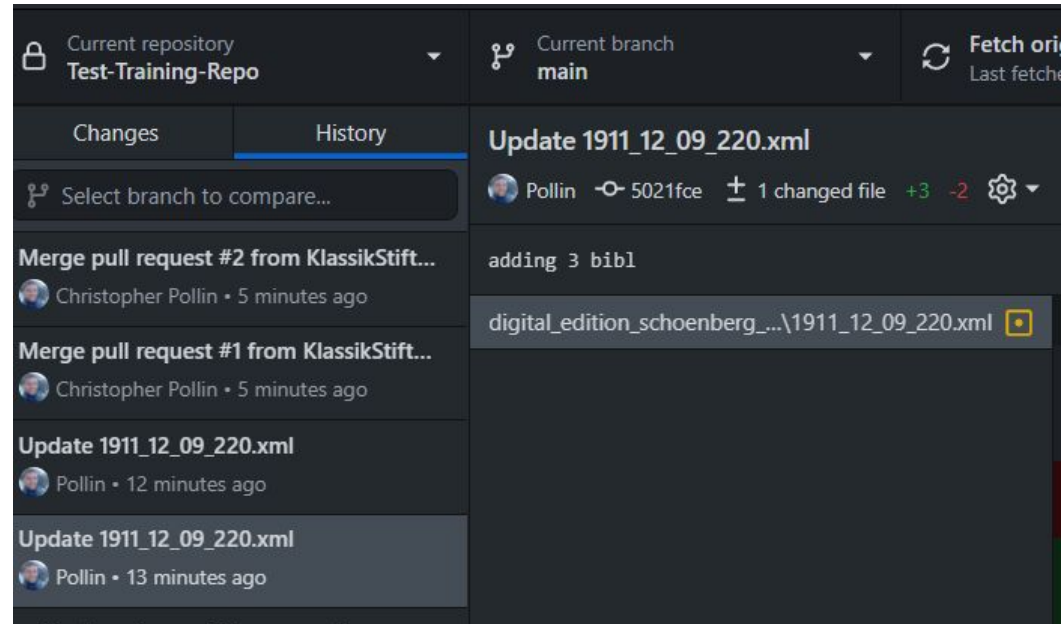


# Branches



# Branches und Pull request

- 3 <bibl> in einer Datei im  
teiHeader branch hinzugefügt
- 1 fehlendes <p> im teiBody  
branch
- merge pull request 1 → 3 <bibl>  
aus teiHeader branch nach main
- merge pull request 2 → 1 <p>  
aus teiBody branch nach main



## Hands-On: eigener Branch

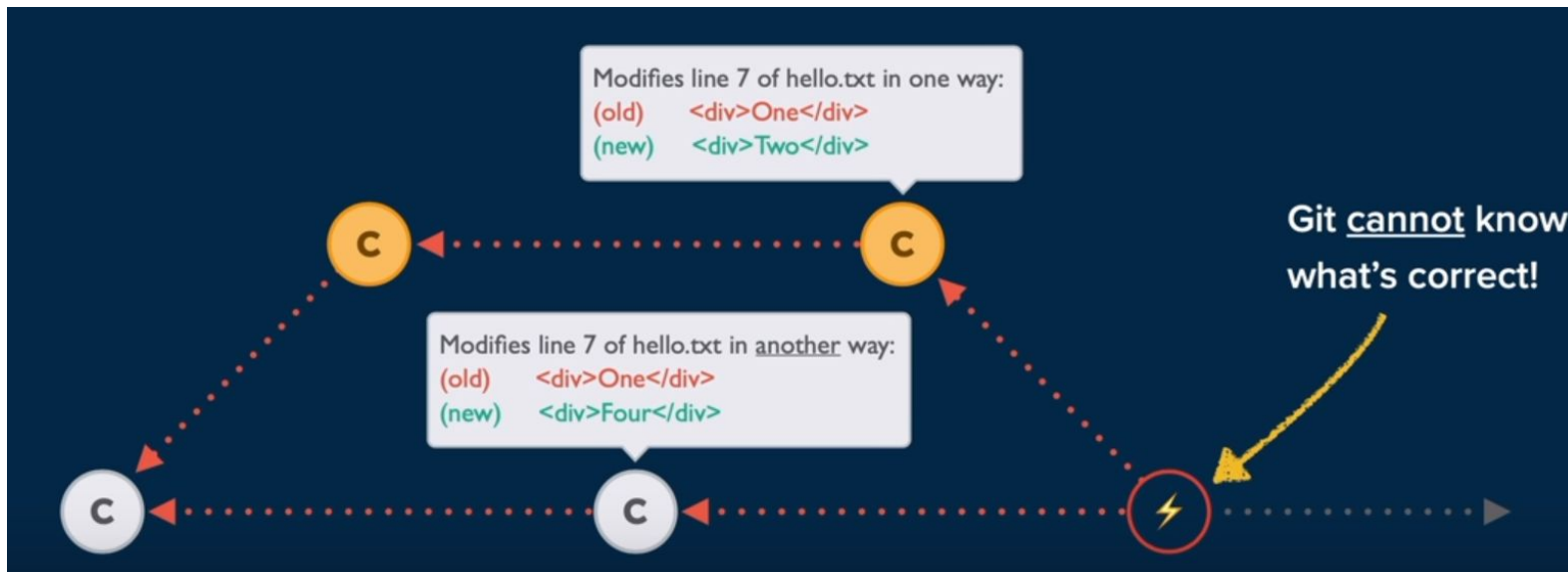
Erzeugen Sie einen neuen Branch mit Ihrem Namen (z.B.: “christopher”).

Nehmen Sie eine Änderung im XML/TEI vor:

[https://github.com/KlassikStiftungWeimar/Test-Training-Repo/tree/main/digital\\_edition\\_schoenberg\\_example/sample\\_letter](https://github.com/KlassikStiftungWeimar/Test-Training-Repo/tree/main/digital_edition_schoenberg_example/sample_letter)

Committen Sie “lustig drauflos”, damit jede/r einen Konflikt bekommt, den wir dann bereinigen (oder auch nicht) bereinigen können.

# Merge Conflicts



- About merge conflicts,  
<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflicts>
- Git for Professionals Tutorial - Tools & Concepts for Mastering Version Control with Git, [https://www.youtube.com/watch?v=Uszj\\_k0DGsg](https://www.youtube.com/watch?v=Uszj_k0DGsg)

# Merge Conflicts

```
14 ▼      <ul>
15 <<<<<< HEAD
16 ▼      <li><a href="index.html">Home</a></li>
17 ▼      <li><a href="about.html">About Us</a></li>
18 ▼      <li><a href="product.html">Product</a></li>
19 ▼      <li><a href="imprint.html">Imprint</a></li>
20 =====
21 >>>>>> develop
22 ▲      </ul>
```

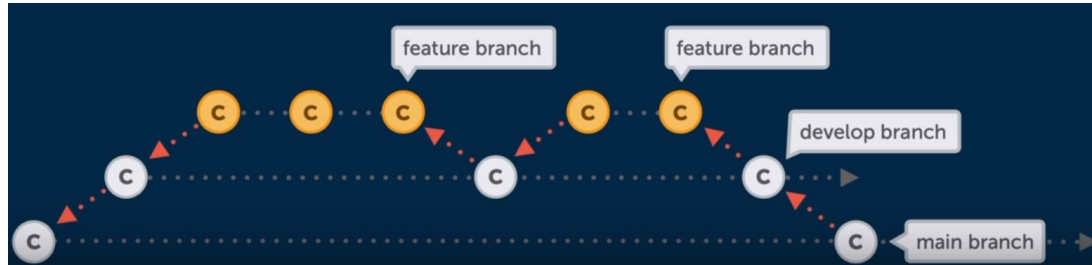
- About merge conflicts,  
<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflicts>
- Git for Professionals Tutorial - Tools & Concepts for Mastering Version Control with Git, [https://www.youtube.com/watch?v=Uszj\\_k0DGsg](https://www.youtube.com/watch?v=Uszj_k0DGsg)

# “Branching Strategies”

## Mainline Development

- Immer integrierend sein → achten Sie darauf, dass Sie Ihre Inhalte gut, zügig und ordentlich mit der Arbeit Ihrer Kolleginnen/Kollegen zusammenführen.
- Wenig Branches
- Kleine Commits

## State, Releases, and Feature Branches



# Kritik an Git



**Hadi Hariri**

@hhariri



Git is not a success story. Git is a failure as a system with a crap user experience that forces you to learn more about the tool you're using that about getting your work done.

2:20 PM · Oct 8, 2020



1.8K



626 people are Tweeting about this

Blogpost:

<https://hadihariri.com/2020/10/14/a-few-words-on-git/>

## Dinge, die wir nicht besprochen haben...

- **\$ git init**  
Um einen lokalen Ordner zu einem Git Repository zu machen.
- **Fork a repo**  
Eine eigene Kopie eines existierenden Repos, um es selbst weiter zu bearbeiten.
- **GitHub Actions**
- **weitere Tools (z.B. GitKraken)**



# Git bzw. GitHub installieren

## Distribution Ubuntu

Paketverwaltungstool: apt  
\$ apt-get install git

## macOS

Xcode Kommandozeilen-Tools  
\$ git --version

## Windows

Binärer Installationsassistent, <https://git-scm.com/download/win>

Oder GitHub Desktop <https://desktop.github.com/>