

Arbeiten mit Github

Moderne Software-Entwicklung erfordert zwingend den Einsatz eines Source-Code-Management-Systems, so wie GIT. Das ist für eine Team-Arbeit sehr wichtig.

1. Theorie
2. VS-Code und ein vorhandenes Projekt mit github.com verbinden
 - a. Übung: Projekt hochladen (VS-Code mit github.com verbinden) – repository automatisiert anlegen lassen
 - b. Wenn man das Repository eigenhändig erstellen möchte
3. tägliche Arbeiten
 - a. commit - Änderungen im lokalen VS-Code lokal für github.com oder desktop.github bereitstellen
4. Issues verwenden = ToDo-Liste
5. Kollegen einladen und gemeinsam arbeiten
6. Umgekehrte Reihenfolge - Clonen von GitHub auf deinen PC in VisualStudio Code
7. Info zu github.com – löschen von repositories

1.)Theorie

1. **nach allen Einrichtungen endlich täglich arbeiten**
 - a. Commit – Änderungen von VS-Code an github.com oder desktop.github senden
 - b. push origin - Änderungen von desktop.github an GITHUB senden
 - c. Issues nutzen
2. Collaborateur einladen mitzuarbeiten
3. Umgekehrte Reihenfolge - Clonen von GitHub auf deinen PC in VisualStudio Code
4. Info zu github.com – löschen von repositories

Vorgang:

Wenn Git und auch Github.Desktop installiert und eingerichtet sind und auch ein Konto bei github.com (noch ohne repository) besteht:

- Erste Speicherung mittels Verbindung in VisualStudioCode

Laufende Änderungen:

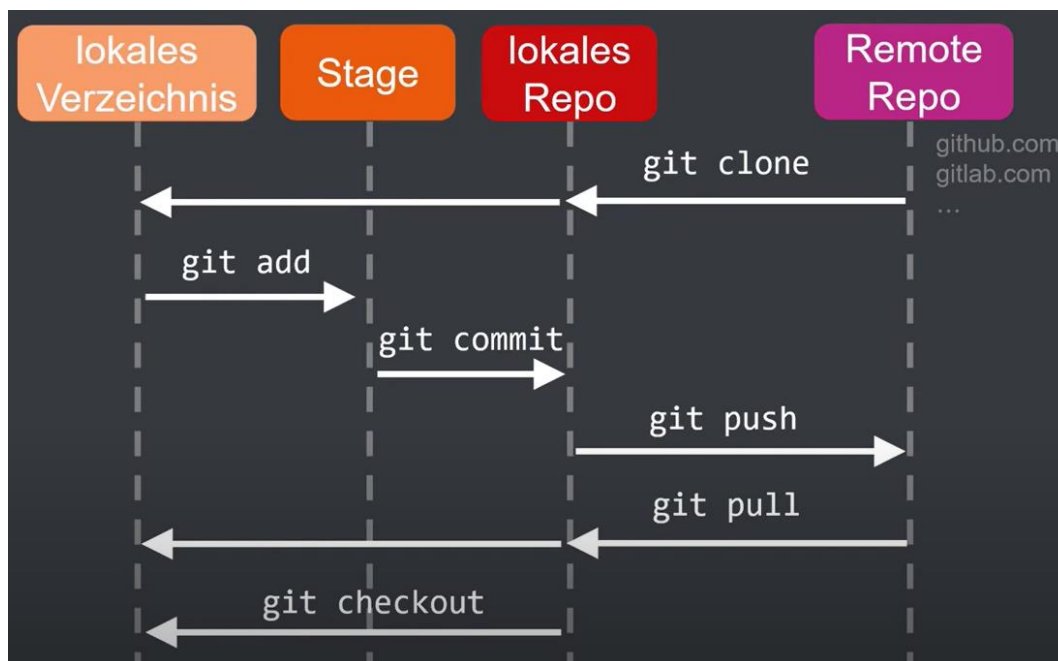
- Anlegen von Ordner und Datei in Visual Studio Code
- Verbindung in VS-Code hat nach dem Speichern ein blaues Symbol beim Icon „Quellcodeverwaltung“
- Dort Kommentar eingeben und Haken klicken
- Dann wird die Änderung in Github.Desktop sichtbar und man kann dort „push origin“ klicken
- Nun wird es auf das Repository auf github.com geclont.

Es gibt 5 (6) Befehle:

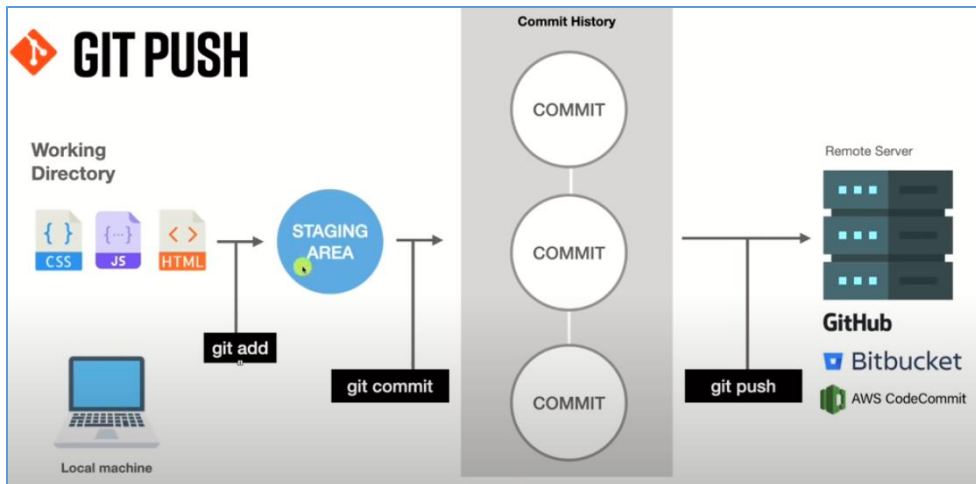
- **git clone:** vom Remote Repo (z.B. github.com) auf das lokale Verzeichnis. Dieser Befehl wirkt so, dass es zuerst in das lokale Repo überführt wird und von da in das lokale Verzeichnis. Dort kann man dann so damit arbeiten, als wäre es eine lokale Datei. Dort kann man es auch löschen.
- **git add:** damit werden Änderungen auf die "stage". Diese Änderungen sind noch nicht „committed“.
- **git commit:** hier wird auch eine „message“ genutzt. Üblicherweise wird ein commit **am Ende des Tages oder wenn man eine größere Menge an Änderungen abgeschlossen hat**, durchgeführt. Damit ist der Code im lokalen Repository.
- **git push:** erst hiermit wird der Code vom lokalen Repo in das Remote Repo (z.B. github.com) übertragen.
- **git pull:** damit werden Änderungen von KollegInnen in mein lokales Repo überführt. Das ist so ähnlich wie „git clone“, nur dass Aktualisierungen durchgeführt werden.
- **git checkout:** (nicht so wichtig für uns) damit kann man unter mehreren „branches“ unterscheiden, womit man verschiedene Varianten der gleichen Software in den verschiedenen „branches“ verwalten kann.

pull != clonen

Das Verzeichnis ist immer der aktuelle Schnappschuss, auf dem man gerade arbeitet.



<https://www.youtube.com/watch?v=TpwWq5xigSo>



Erklärung der Grafik:

Der linke Bereich wird durch GIT am eigenen PC abgedeckt. Interessant ist vielmehr der Bereich von den commits zum gehosteten GITHUB. Dort soll unser Projekt hin und mit den anderen Teilnehmern gemeinsam entwickelt werden.

<https://www.youtube.com/watch?v=mVnZVw4KJnc>

2)VS-Code und ein vorhandenes Projekt mit github.com verbinden

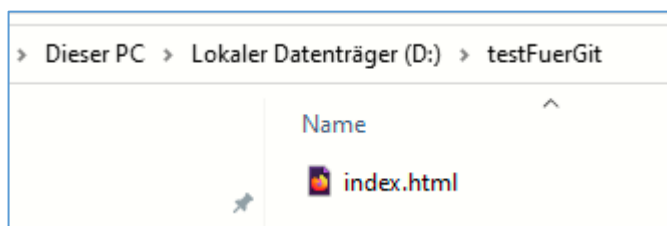
2a)Übung:

Projekt hochladen (VS-Code mit github.com verbinden) – repository automatisiert anlegen lassen

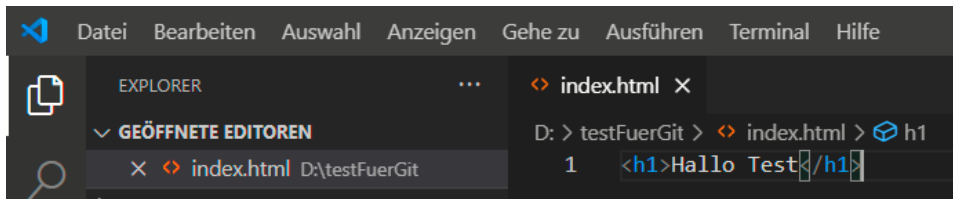
Ausgangssituation: es ist noch kein entsprechendes repository vorweg in github.com angelegt worden, sondern das wird im Zuge des Hochladens erstellt.

Vorbereitung:

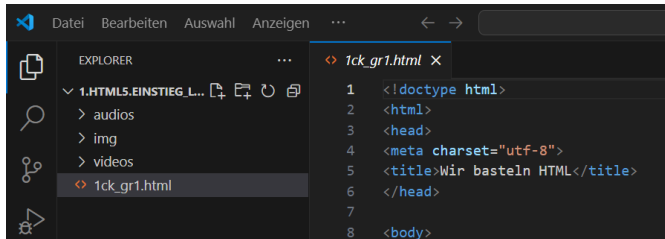
- a)Öffne in VS-Code ein älteres Projekt von dir, das eine index.html enthält oder
- b)erstelle ein neues Projekt mit einer „index.html“ und speichere es auf dem Laufwerk im Ordner „testfuerGit“.



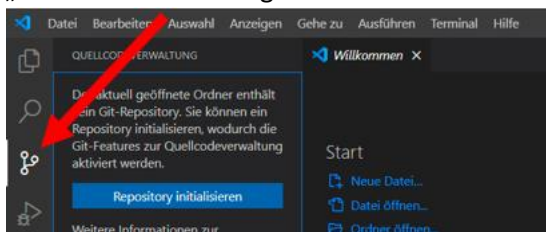
Öffne VS-Code und öffne den gerade angelegten Ordner.



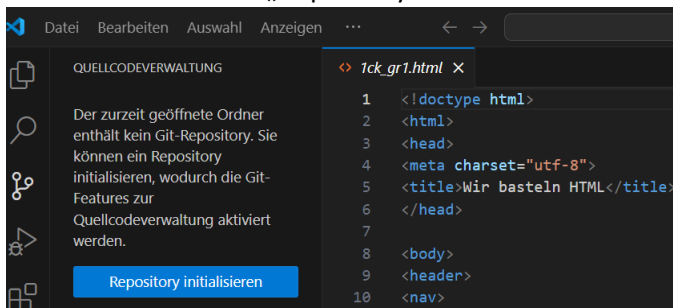
1. Projekt in VisualStudio-Code anzeigen



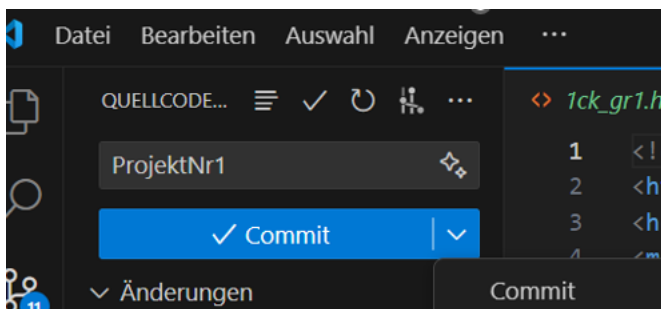
2. Klicke auf das 3. Symbol links in der Leiste (Quellcodeverwaltung). Nun findet man im Icon „Quellcodeverwaltung“ die Bereitschaft für GIT.



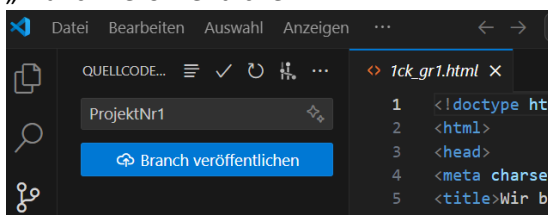
3. Klicke auf den Button „Repository initialisieren“



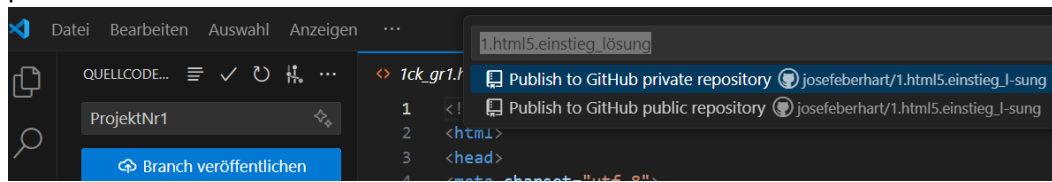
4. Gib einen Text ein und dann auf den Pfeil nach unten beim Button „Commit“ .



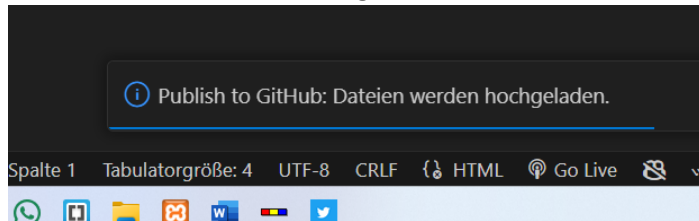
5. Zum Upload der Files benötigt man commit und push. Daher wird als nächstes der Button gezeigt „Branch veröffentlichen“.



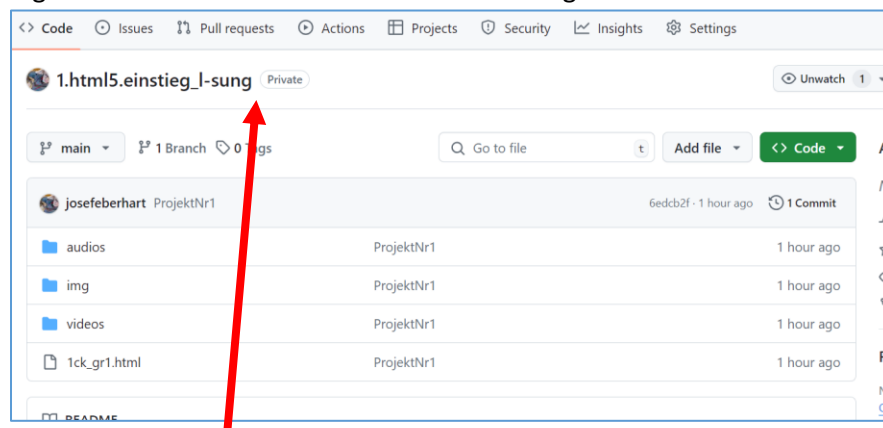
6. Wähle aus, ob es privat oder public ist in der nun angezeigten Auswahl. Es soll hier zum Testen privat bleiben.



7. Dann werden die Daten hochgeladen.



8. Ergebnis: ist nun vorhanden und kann laufend geändert usw. werden

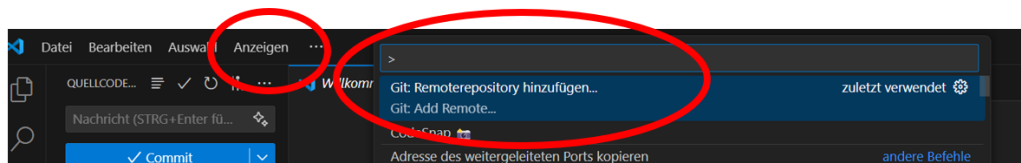


Beachte: der Name wurde übernommen und ist mit Umlauten ein Problem. Daher: vorher den Namen optimieren.

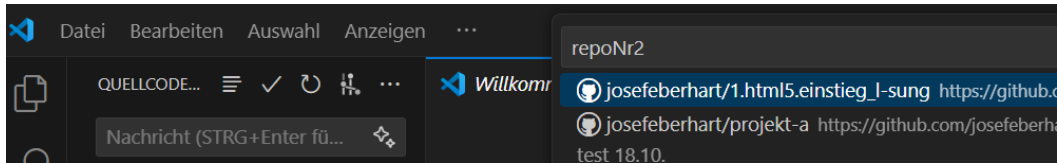
<https://www.youtube.com/watch?v=3Tn58KQvWtU>

2b) Wenn man das Repository eigenhändig erstellen möchte:

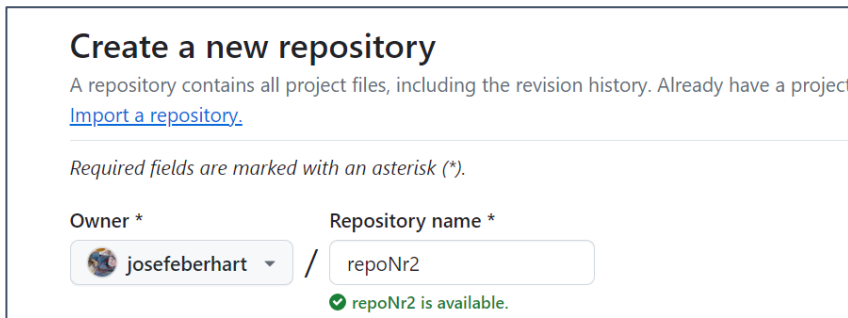
- Bevor man „commit“ klickt
- Menü Anzeigen
- Befehlspalette
- Wenn nicht vorhanden, dann gib ein „git add re...“, damit es angezeigt wird



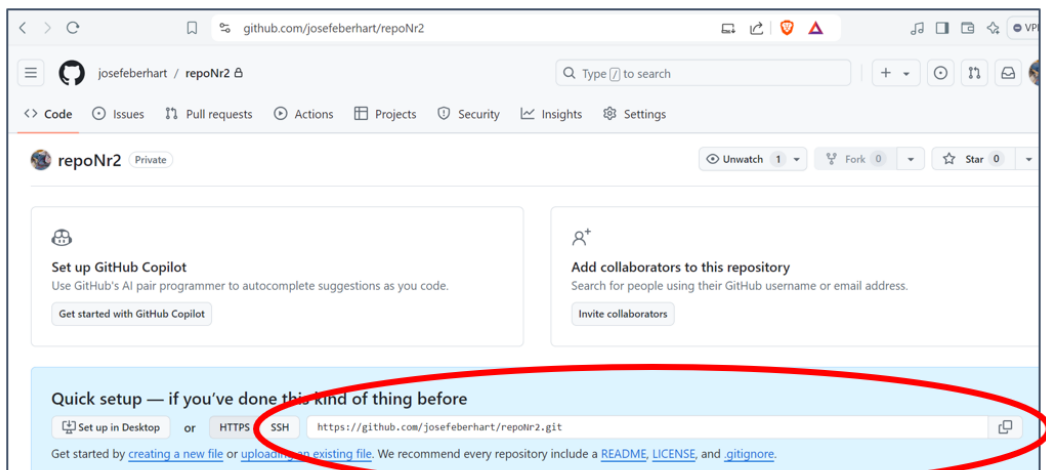
- Gib dann einen neuen Namen ein, weil ja ein neues Repository angelegt werden soll



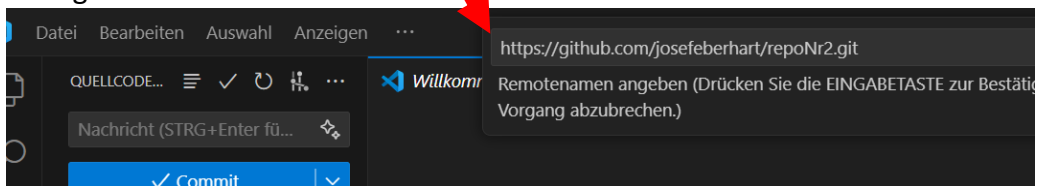
- Dann die URL aus deinem github.com: öffne github.com und erstelle manuell ein neues repository, hier wieder „privat“



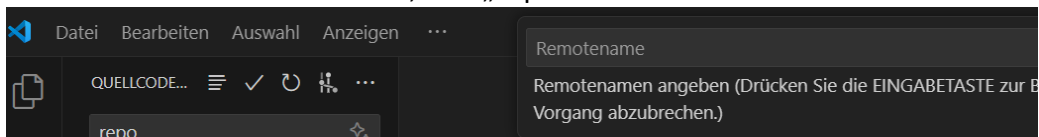
Kopiere dann den Link und füge ihn in VS-Code ein:



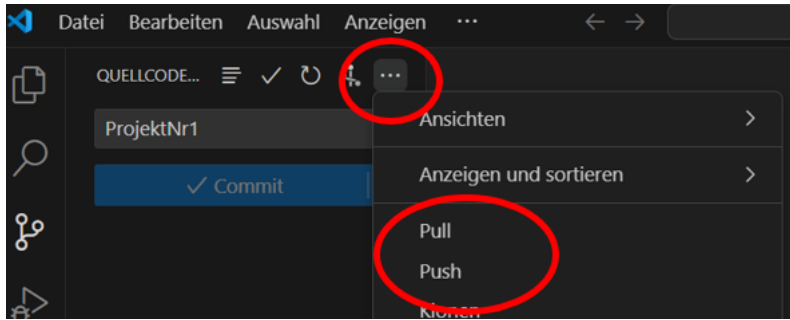
- Einfügen und Enter



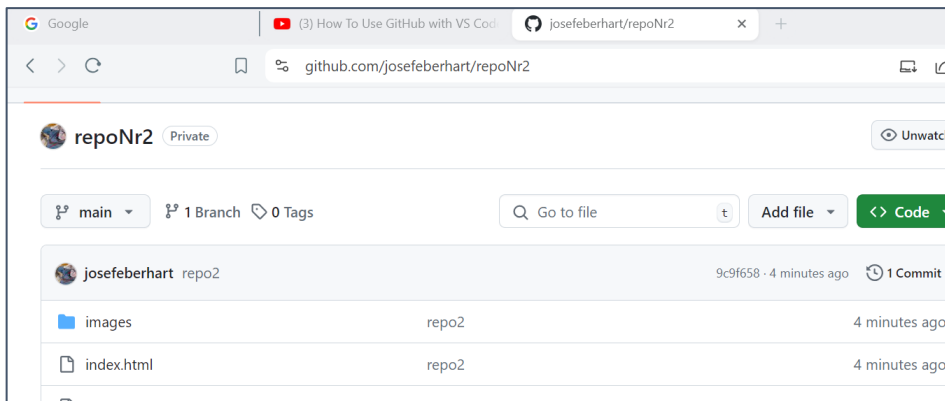
- Dann noch einen Remotenamen, hier „repo2“



- Mit den 3 Punkten mehr Inhalt anzeigen lassen und nutze hier „push“



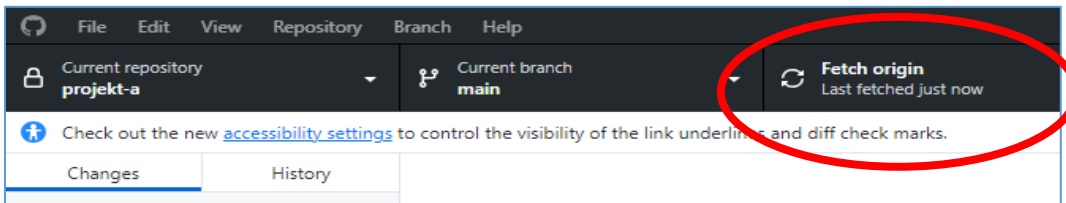
Ergebnis:



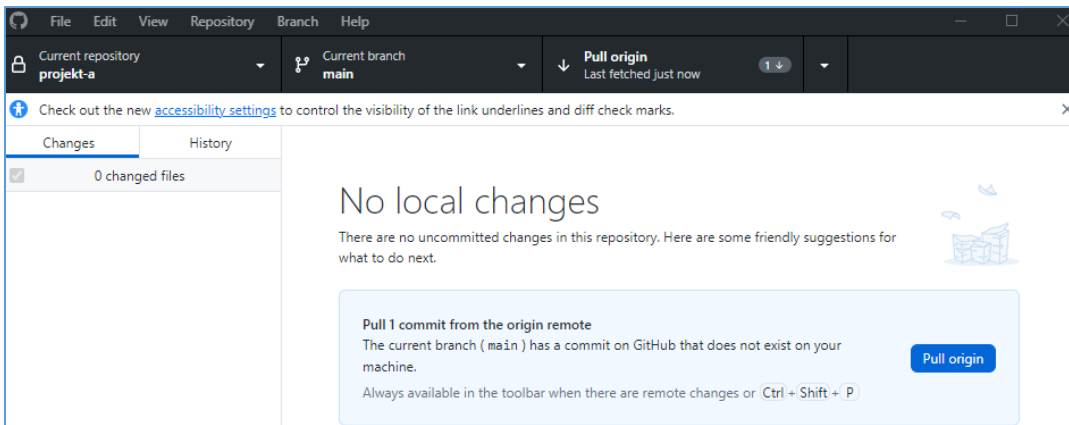
3)tägliche Arbeiten:

desktop.github:

Bei Arbeitsbeginn in der Früh kann man in „desktop.github“ mit „fetch origin“ mögliche Änderungen der KollegInnen auf seinen Rechner holen (pull), damit man aktuell weiterarbeiten kann.



Wenn es Änderungen gab, dann wird das Fenster „Pull origin“ angezeigt, das dann zum Kicken ist.

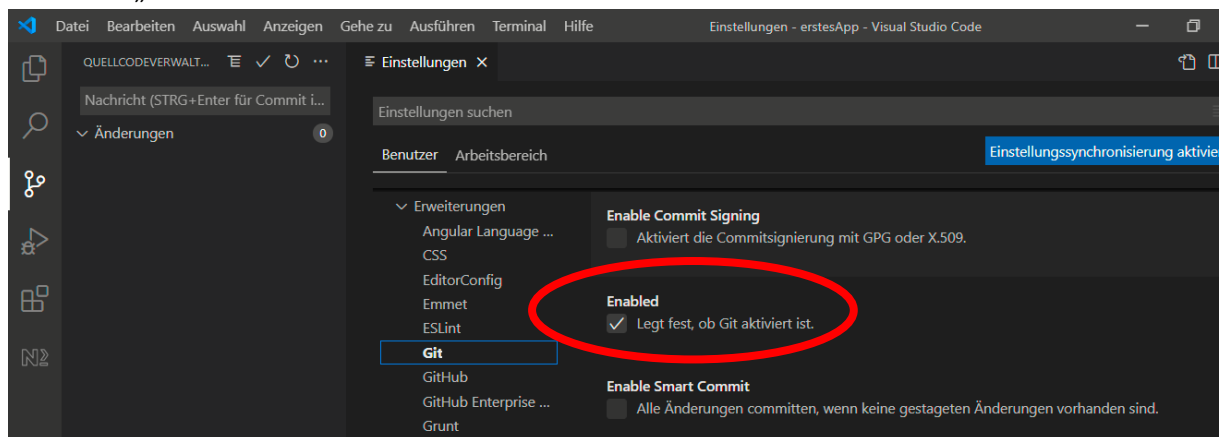


3a)commit - Änderungen im lokalen VS-Code lokal für github.com oder desktop.github bereitstellen

<https://www.youtube.com/watch?v=3Tn58KQvWtU>

Öffne das Projekt in VS-Code.

Kurzer Test: ob GIT in den Einstellungen (ganz unten links) aktiviert ist - bei Benutzer, Erweiterungen Haken bei „Git – enabled“.



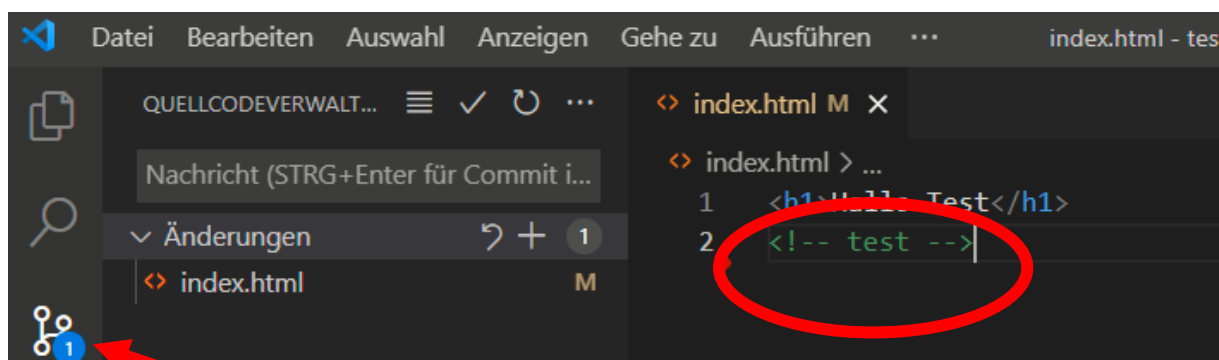
Upload der Files auf Desktop.Github

Das nennt man „Commit“.

Da mit Github-Desktop das Projekt schon auf GITHUB hochgeladen wurde, muss man es nicht auch hier machen.

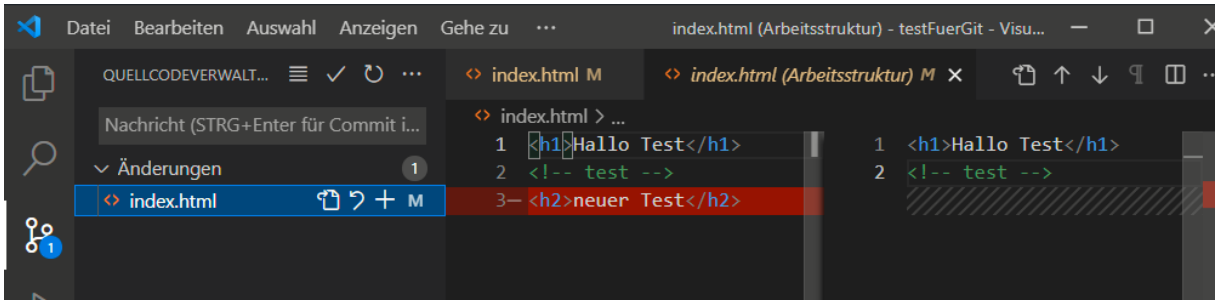
Übung:

erstelle in einem vorhandenen File einen Kommentar (am besten mit der STRG-Taste und Raute für die automatische Darstellung der Kommentarzeichen) und dann den Text „test“.



Dafür kann man aber sehr bequem die Änderungen verwalten: **wenn eine Änderung mit „Speichern“ abgeschlossen ist, wird diese erkannt und sofort dies auch angezeigt** – siehe blaues Symbol in der Leiste links:

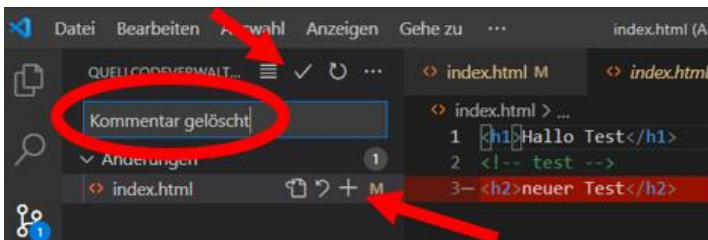
Klicke darauf und dann nochmals auf die Datei unterhalb von „Änderungen“, hier auf „index.html“ dann man sieht die Darstellung in zwei Fenstern rechts davon.



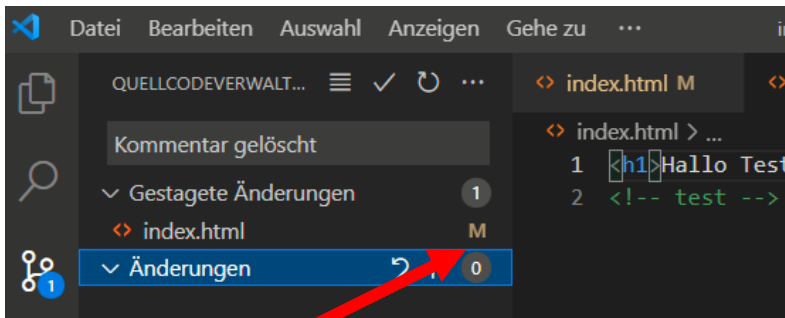
Klickt man im Code auf den veränderten Code, sieht man auch **WER** die Veränderung durchgeführt hat.

COMMIT:

Dafür **gib einen Text** ein, der die Änderung beschreibt und klicke auf den Haken damit diese Änderung übermittelt wird zum lokalen GIT-repository auf dem PC (commit)

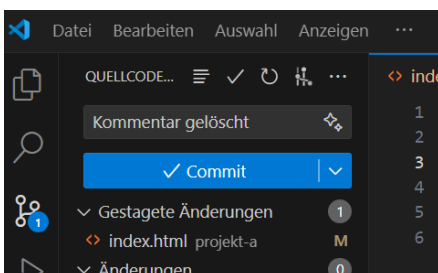


- Dann auf das „+“-Symbol, um die Änderungen zu „**stagen**“. Das ist soviel wie „git add“, damit werden Änderungen auf die „stage“ geschoben. Diese Änderungen sind noch nicht „committed“.



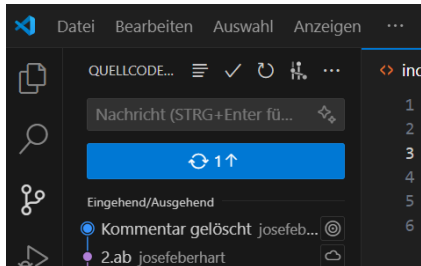
Das M bedeutet „modified“.

Klicke auf den „Commit“-Button.



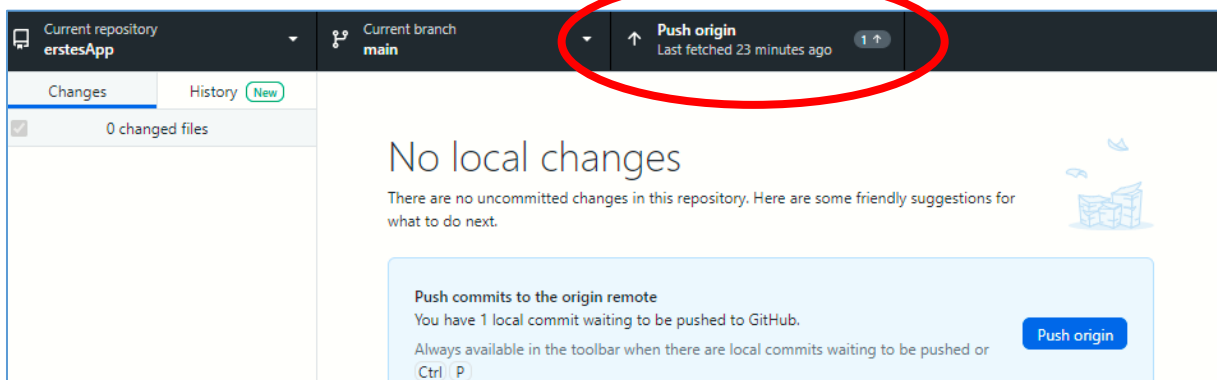
- Variante 1: **ohne desktop.github**
Danach auf „sync“ bzw. auf den blauen Button, um mit dem Server sich zu verbinden.
Damit umgeht man den Desktop.Github, weil es gleich auf den Server landet.

<https://www.youtube.com/watch?v=0jzjz4MZ4ZU>



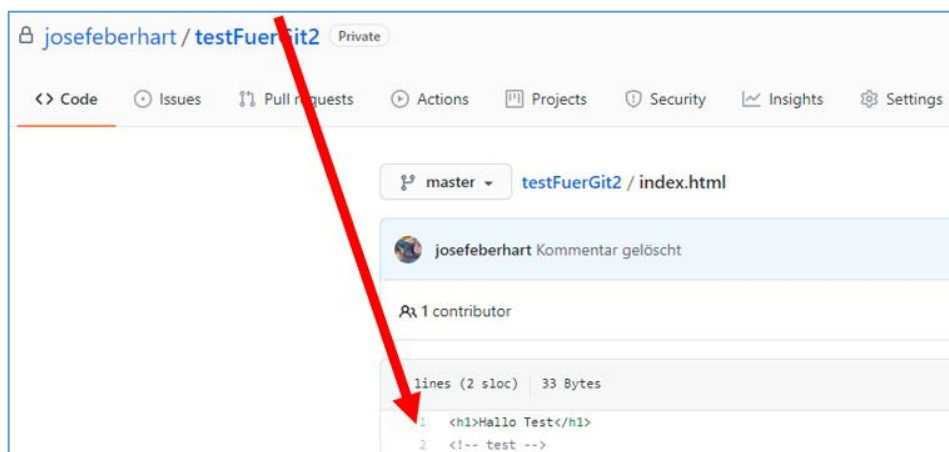
- Variante 2: **über desktop.github arbeiten**
Man kann aber nach dem „Commit“ nicht auf Synchronisieren klicken, sondern in Desktop.Github gehen und dort „Push origin“ klicken. Wechsle daraufhin zu **zu Desktop.Github.**

Es wird in Github-Desktop als „Push“ verfügbar gemacht. „You have 1 local commit waiting...“



Klicke darauf und schiebe es somit auf github.com

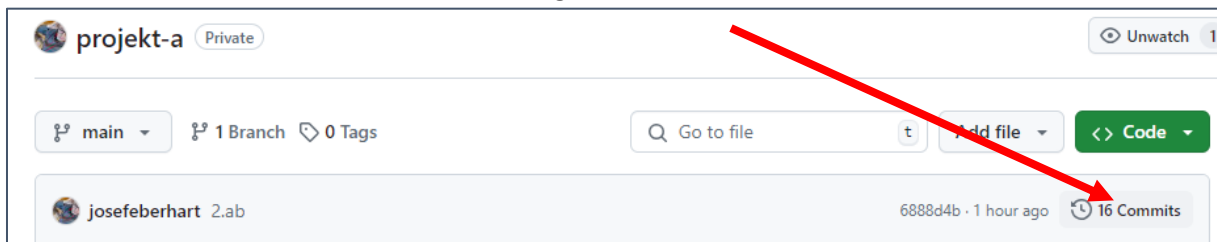
Danach wird es automatisch in das Repository auf www.github.com übermittelt: nach einem Refresh ist es verfügbar



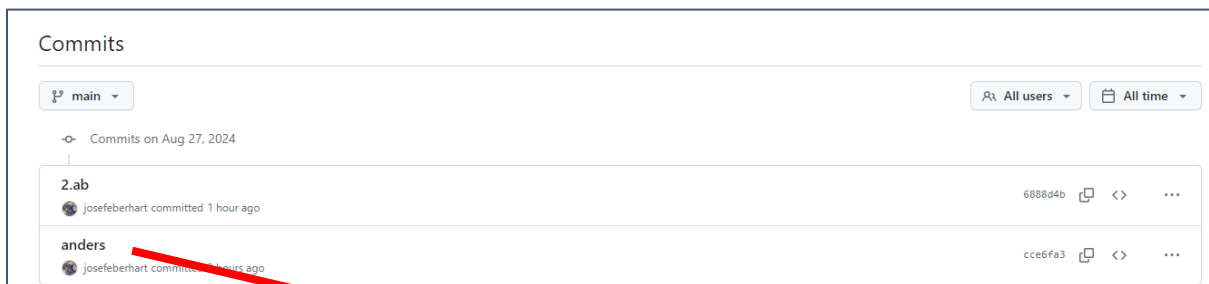
https://www.youtube.com/watch?v=3fUbBnN_H2c ab 1.54.30

In github.com sieht man in seinem Repository auch die Anzahl der vorgenommenen „commits“.

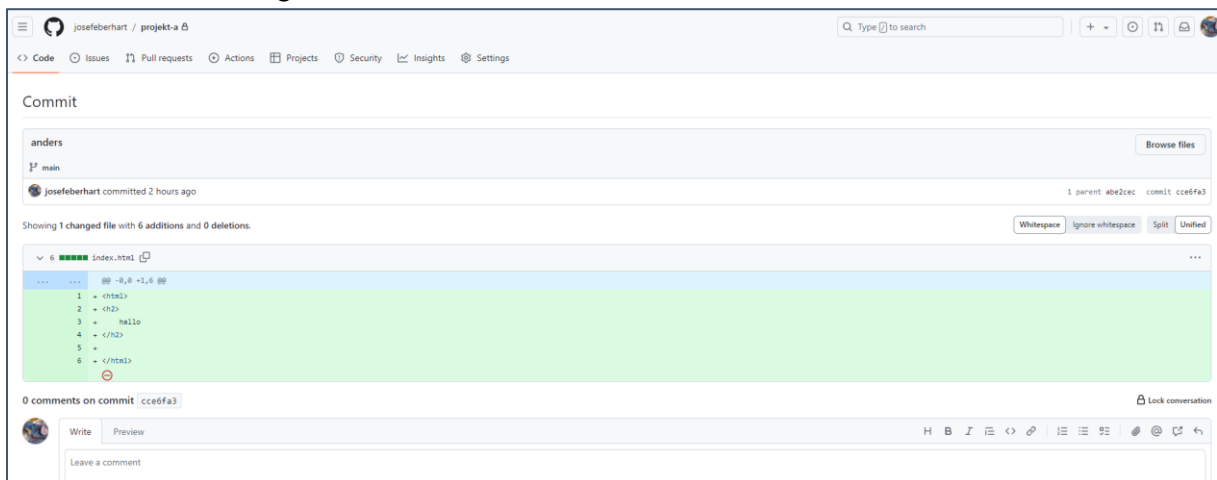
Klickt man auf „commits“ dann sieht man alle genau, wie auf einem „Zeitstrahl“.



Inhalt:



Klickt man auf die Überschrift „message“ hier z.B. auf „anders“ sieht man die Änderungen, welche Dateien und Code sich geändert haben.

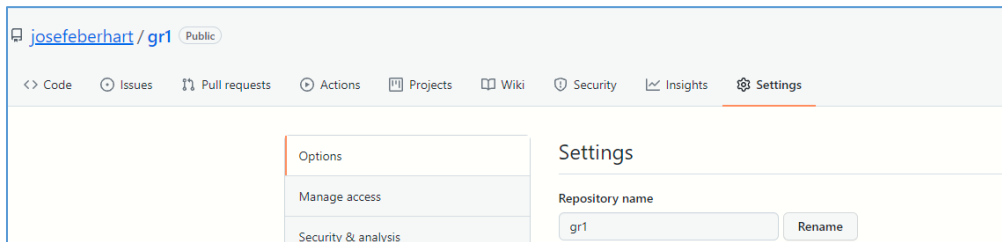


4)Issues verwenden = ToDo-Liste

<https://www.youtube.com/watch?v=0jzj4MZ4ZU> ab 17:30

5)Kollegen einladen und gemeinsam arbeiten

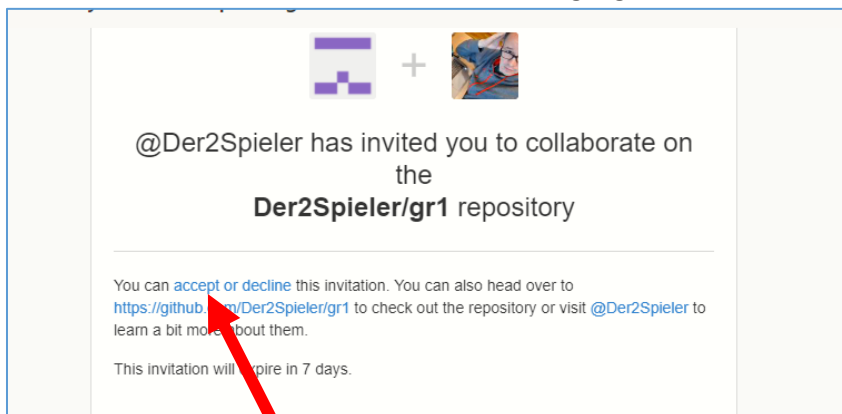
In den „settings“ auf „Manage access“ klicken



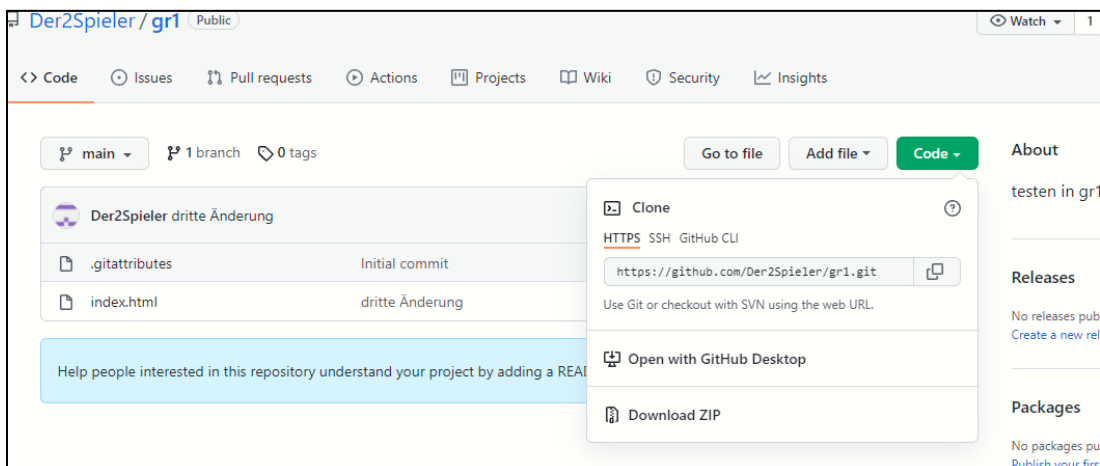
Dann auf „add“ und

- eine E-Mail Adresse des Kollegen/in eingeben
- oder einen bestehenden Namen auf github.com suchen und wählen,

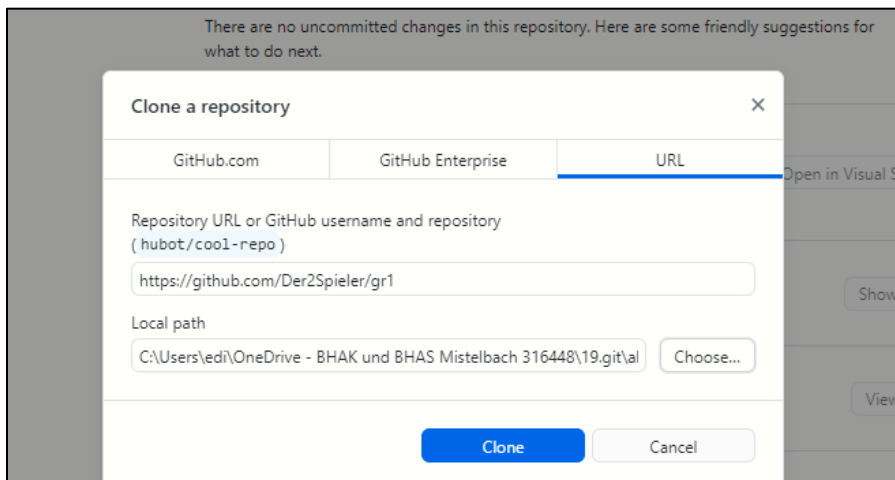
In beiden Fällen erhält er in einer E-Mail den Zugang:



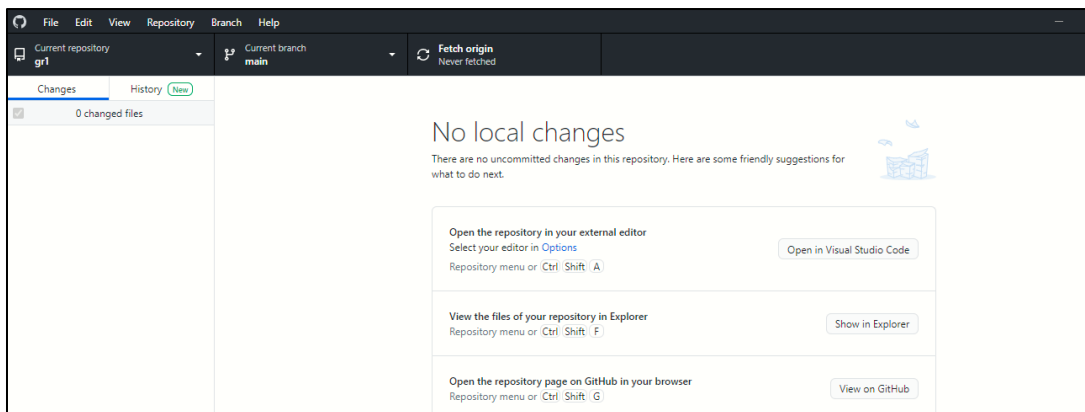
Klicke im Mail auf „accept“ und dann öffnet sich dessen Github.com repository, wo man den grünen Button „Code“ klickt und dort auf „open with Github Desktop“



Dann wähle den „local path“, der aber nicht in einem eigenen Repository auf dem eigenen PC liegen darf, sondern einen anderen guten Platz finden soll:



Dann öffne es mit Klick auf „Open in Visual Studio Code“:



6)Umgekehrte Reihenfolge - Clonen von GitHub auf deinen PC in VisualStudio Code

Auch ohne Konto-Anmeldung möglich.

Dann ist es lokal auf deinem PC. Änderungen können dann aber nicht wieder zurück hochgeladen werden, weil man ja dort keine Schreibrechte hat. Das ist gut so, man will sich ja nur was abholen zum Testen oder Nutzen. Der „Geclonte“ ist ja nicht daran interessiert, dass irgendwer irgendwas bei ihm ändert.

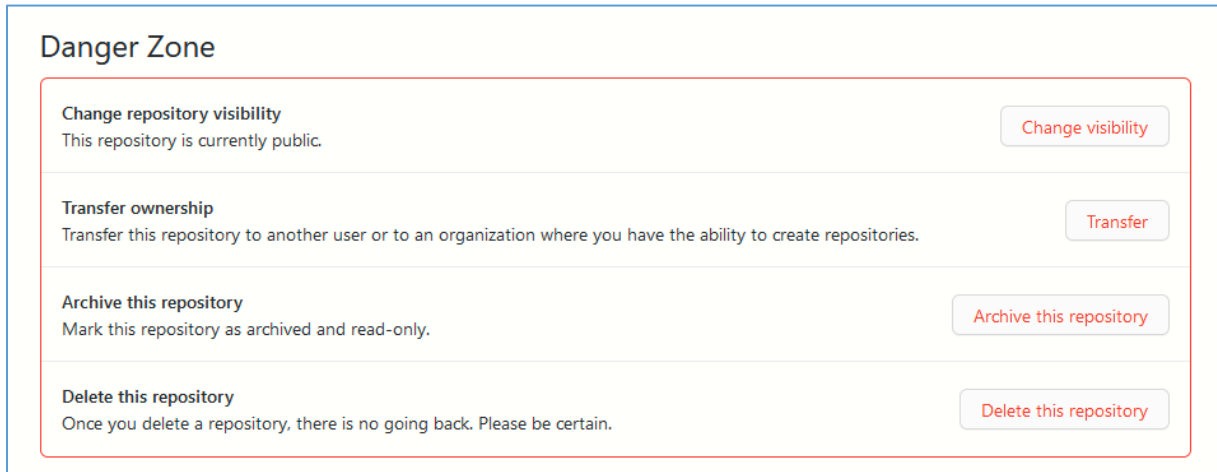
https://www.youtube.com/watch?v=bZZwclXYhc8&list=PLP2TrPpx5VNILOYo5pook-0_0Uy0YLdHW&index=2

<https://www.youtube.com/watch?v=mVnZVw4KJnc> ab ca. 10 Min. 50 Sek.

7)Info zu github.com – löschen von repositories

Löschen von repositories, Umbenennen, Umstellen von privat auf öffentlich.....

Delete in „Settings“ ganz unten



Weitere gute Info:

Ohne Desktop.git.com: https://www.youtube.com/watch?v=lvF_9pf5p3Q

<https://www.youtube.com/watch?v=TpwWq5xigSo> Git Basics, 5 Befehle reichen

pull != clonen

Gutes Plugin in VS-Code für Git:

