

Verwendung von git in Visual Studio Code

1. Voraussetzung: Installation von git
Überprüfen im Terminal / Bash mit `git --version`
2. Oft hilfreich und wichtig, sobald mehrere Personen zusammenarbeiten:
git Konfiguration überprüfen bzw. setzen (--global heißt für alle Projekte)

```
git config --global user.name
```

```
git config --global user.email
```

bzw. (du solltest natürlich deinen Benutzernamen / eMail verwenden!!)

```
git config --global user.name "llinder31"
```

```
git config --global user.email linder.lydia@borgnonnal.at
```

3. Neuer Ordner zum Ausprobieren erstellen (z.B. Homepage2024_deinNAME)
4. Öffne Visual Studio Code mit diesem Ordner - kannst du das noch über d. Windows Kommandozeile?
5. Lege neue Datei an, z.B. index.html



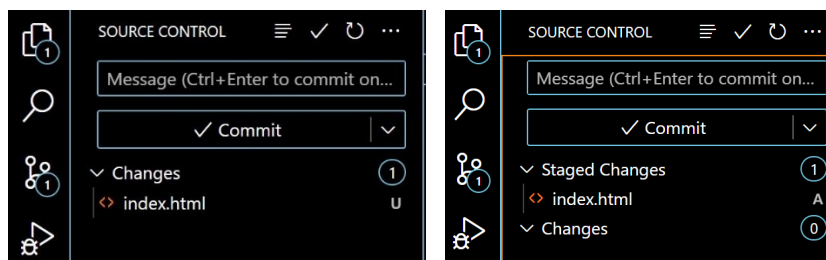
6. Du möchtest die aktuelle Version / diesen Snapshot speichern, schau dazu im Bereich „Source Control“:

- a) Git Repository im Source Control Bereich f. dieses Projekt anlegen mit "italize Rep." (statt git init) – das funktioniert jedenfalls beim ersten Mal.

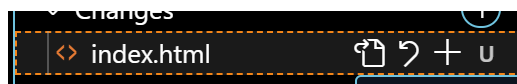
Alternativ gehe im Menü auf View / Command Palette / git clone eingeben (URL auswählen oder eingeben)

Anschließend im Windows Explorer den Speicherort des neuen Repositories wählen u. dann ggf. in neuem Fenster öffnen.

- b) Erstelle eine Datei, z.B. index.html, falls du das noch nicht getan hast:
Git hat gemerkt, dass eine Datei geändert wurde, die Zahl der veränderten Dateien wird beim Source Control Symbol angezeigt. Weiters wird mit "U" neben Dateinamen gezeigt, dass diese Datei untracked ist.



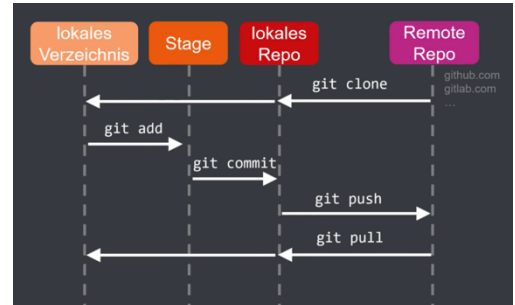
7. Source Control Bereich: add = stagen
Gehe in den SourceControl Bereich, klicke auf das „+“ wenn du eine Datei adden willst.
(Mit “-“ wiederum rückgängig machen)



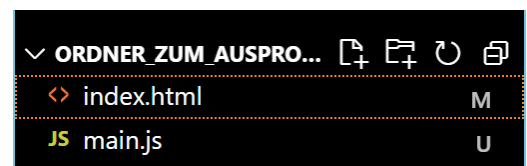
8. Committen ist nur für Dateien aus der Staging Area möglich:
 Klicke dafür das "+" neben dem Dateinamen (erscheint, wenn man mit der Maus drüber fährt)
 Rückgängig machen mit -

Die Commit Message soll kurz, aber aussagekräftig sein!
 Abschicken mit STRG+ENTER oder auf Commit klicken

9. Erst wenn du im Source Control Bereich bei den drei Punkten „push“ auswählst, werden die Änderungen zum GitHub Server übertragen. (Solange du in VSC arbeitest, sind die Daten auch nur dort und nicht in der Bash sichtbar. Dafür musst du die Dateien erst hochladen und danach wiederum mit pull synchronisieren. Probiere das aus! Sieh dir den Ablauf in der folgenden Grafik an.)



10. Neue Datei, z.B. js-Datei; in index.html referenzieren (<script>)
 M .. Steht für modified
 U .. Steht für untracked



11. Erstelle eine Datei `style.css` und füge dies im Header ein (link rel...
 Probieren commit ohne staging (also add)
 Geht auch, Kommentar eingeben; Meldung im Fenster bestätigen (add wird mit commit gemacht)

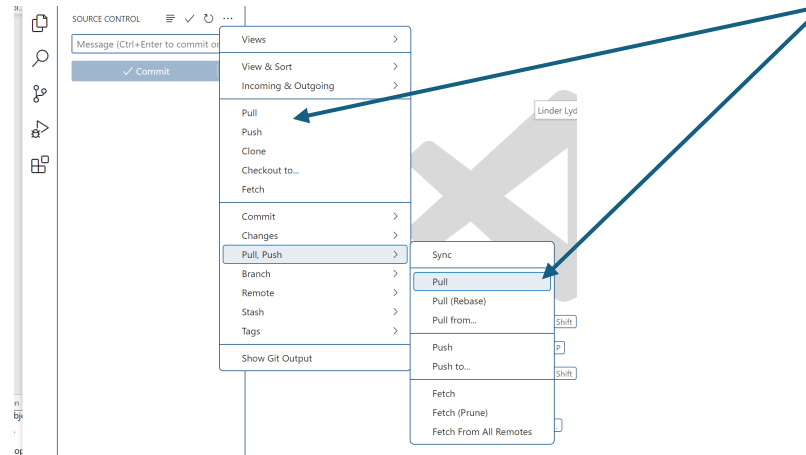
Mache Änderungen in `styles.css` (z.B. wie unten in der Beispielsdatei) und committe diese Änderungen. Probiere Daten zu ändern, ergänzen und zu löschen und Änderungen wiederum in deinem Repository Hochzuladen. Wie werden diese Änderungen (Erweiterungen, Löschen von Zeilen, Textänderungen) in hervorgehoben?

Lösung der Datei: index.html

Index.html fertig	
<pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <meta charset="UTF-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1.0"> 6 <title>Document</title> 7 <link rel="stylesheet" href="style.css"> 8 9 </head> 10 <body> 11 Hallo 12 <script src="./main.js"></script> 13 </body> 14 </html> 15 </pre>	

Style.css fertig	
<pre> body{ font-family: Arial, Helvetica, sans-serif; color: #222; font-size: 1.2rem; } </pre>	

Verwende zum Synchronisieren deines lokalen Repositorys mit dem online Repository das Menü bei den drei Punkten im Source Control Bereich oder klicke auf den Pfeil rechts neben „Commit“:
Dabei wird dir wahrscheinlich empfohlen, ob dieser Abgleich mit GitHub automatisch regelmäßig gemacht werden soll.



Arbeite nun in unserem gemeinsamen Repository – Abgabe zum Teil in Teams, z.T. im Repository:

- Erstelle im Skriptum ein Unterkapitel zu Versionsverwaltung → Versionsverwaltung mit Visual Studio Code. Halte die wichtigsten Punkte fest! Beschreibe auch, wie du Änderungen erkennst. Mach einen Screenshot zur Abgabe in **Teams**.
- Screenshot des aktuellen Repositorys nach clone bzw. pull
(https://github.com/llinder31/2024_Inf_7D.git)
(Abgabe in **Teams**)
- Lade einen Screenshot deines Inhaltsverzeichnisses der aktuellen Kapitel deines Skriptums im gemeinsamen Repository hoch. (**Abgabe im gemeinsamen Rep.**)
Benenne die Datei aussagekräftig – dein Name muss im Dateinamen enthalten sein. Vergleiche dein Inhaltsverzeichnis mit dem deiner Kolleg:innen. Ist dein Skriptum vollständig?

Zur Wiederholung oder auch Problemen kannst du dir dieses kurze Video (4,5 min) ansehen:

<https://youtu.be/uuNbZ79SkEo>

In einem weiteren Video (siehe Teams, Kanal zur Versionsverwaltung) kannst du bisher Gelerntes zum Thema wiederholen und vertiefen.