

Protokoll

G 5A.3.2: MIT GIT AUF DER KONSOLE ARBEITEN

ANDREAS SÜNDER 3BHIT

Inhalt

1. Erstelle dir einen eigenen Ordner mit dem Namen gitLoginname.	2
2. Erstelle in diesem Ordner ein neues leeres git-Repository	2
3. Erstelle in diesem Ordner eine Datei mit dem Namen Nachname.txt und schreibe den Text Nachname gefolgt von der Ziffer 1 hinein.....	2
4. Überprüfe den Status von git, wobei die in Schritt 3 erstellte Datei noch untracked sein soll.	2
5. Füge diese Datei zum Staging-Bereich hinzu und überprüfe den Status von git.	2
6. Ändere die Ziffer in deiner Datei auf den Wert 2, speichere und überprüfe den Status von git....	3
7. Füge die geänderte Datei zum Staging-Bereich hinzu und überprüfe den Status von git.	3
8. Übertrage alle Dateien im Staging-Bereich als neue Version in das Repository und zeige die Log- Einträge deiner Versionen im Repository an.....	4
9. Füge nun die Datei <i>Jahrgang.info</i> zu deinem Verzeichnis hinzu und schreibe deinen Jahrgang hinein. Überprüfe den Status von git.	4
10. Schließe alle .info-Dateien von der Versionierung aus und überprüfe danach den Status von git	4
11. Füge alle Änderungen als neue Version zu deinem Repository hinzu und zeige die Log-Einträge deiner Versionen im Repository an.....	5
12. Erstelle einen neuen leeren Unterordner mit dem Namen <i>Vorname</i> und überprüfe den Status von git.....	5
13. Füge in den Unterordner eine Datei mit dem Namen <i>Vorname.txt</i> ein, die deinen Vornamen als Text enthält. Ändere in deinem Basis-Ordner die Datei <i>Nachname.txt</i> so dass die Ziffer den Wert 3 hat, speichere und überprüfe den Status von git.....	5
14. Füge alle Dateien mit * zum Staging-Bereich hinzu und überprüfe den Status von git.....	6
15. Erstelle eine neue Version und zeige deinen Versionsverlauf (Log-Einträge) an.....	6
16. Erstelle auf Github ein neues leeres Repository unter eurem Github Account.....	7
17. Füge dieses Github Repository bei deinem lokalen Repository als Remote-Repository unter dem origin hinzu.....	7
18. Übertrage die Versionen von deinem lokalen Repository auf dein Github-Repository und überprüfe dann dort, ob deine Dateien vorhanden sind und ob du auch alle Versionen siehst.....	8

1. Erstelle dir einen eigenen Ordner mit dem Namen gitLoginname.

Dies kann mit dem Befehl `mkdir gitAsuender` erreicht werden.

```
~/mnt/c/Users/andis/Documents/TGM/3BHIT/Lernbüro SEM/6 Sa.3.2: mkdir gitAsuender 08:42:44
```

2. Erstelle in diesem Ordner ein neues leeres git-Repository

Der Befehl `git init` erstellt in dem aktuellen Verzeichnis ein neues lokales Repository und fügt den Unterordner `.git` hinzu.

```
~/mnt/c/Users/andis/Documents/TGM/3BHIT/Lernbüro SEM/6 Sa.3.2/gitAsuender: master 71 git init 08:43:51
```

3. Erstelle in diesem Ordner eine Datei mit dem Namen Nachname.txt und schreibe den Text Nachname gefolgt von der Ziffer 1 hinein.

Unter Linux kann man den Befehl `touch` verwenden, um eine Datei zu erstellen. Bearbeitet wurde diese mit `nano`.

```
~/mnt/c/Users/andis/Documents/TGM/3BHIT/Lernbüro SEM/6 Sa.3.2/gitAsuender: master 71 touch Suender.txt 08:43:51
```

```
GNU nano 4.8 Suender.txt
Suender 1
```

4. Überprüfe den Status von git, wobei die in Schritt 3 erstellte Datei noch untracked sein soll.

Wie hier zu sehen ist, werden erstellte Dateien nicht automatisch zur Versionskontrolle hinzugefügt, weshalb sie unter `git status` rot angezeigt werden.

```
> ls
Suender.txt
```

```
> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Suender.txt

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

5. Füge diese Datei zum Staging-Bereich hinzu und überprüfe den Status von git.

Dies geschieht mit dem Befehl `git add <file>`. `git status` zeigt jetzt den gestagten Bereich an inkl. der hinzugefügten Datei.

```

> git add Suender.txt .
> git status
On branch master

No commits yet

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

```

6. Ändere die Ziffer in deiner Datei auf den Wert 2, speichere und überprüfe den Status von git.

```

GNU nano 4.8 Suender.txt Modified

```

7. Füge die geänderte Datei zum Staging-Bereich hinzu und überprüfe den Status von git.

Die Datei `Suender.txt` ist zwar schon hinzugefügt, jedoch noch nicht commitet. Würde ich dies jetzt tun, so wäre die neue Version der Datei nicht dabei, sie scheint unter `git status` rot. Nach dem Aufruf von `git add Suender.txt` ist dies behoben.

```

> nano Suender.txt
> git status
On branch master

No commits yet

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

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Suender.txt

> git add Suender.txt
> git status
On branch master

No commits yet

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

```

8. Übertrage alle Dateien im Staging-Bereich als neue Version in das Repository und zeige die Log-Einträge deiner Versionen im Repository an

In der Kommandozeile commitet man alle Changes im Stage-Bereich mittels `git commit -m "<message">`. Lässt man die letzten zwei Argumente aus, so öffnet sich der Standard-Editor von `git` (etwa `nano`, `vscod` etc.), in welchem man die Commit-Message einfügen muss.

```
> git commit -m "Add Suender.txt"
[master (root-commit) 4686569] Add Suender.txt
1 file changed, 1 insertion(+)
create mode 100644 Suender.txt
```

```
commit 4686569dc35d1a9c9f9e2e810f96573a4461e7de (HEAD -> master)
Author: Andreas Suender <asuender@student.tgm.ac.at>
Date:   Fri Nov 12 12:11:23 2021 +0100

    Add Suender.txt
(END)
```

9. Füge nun die Datei *Jahrgang.info* zu deinem Verzeichnis hinzu und schreibe deinen Jahrgang hinein. Überprüfe den Status von `git`.

Die Datei erscheint rot, nachdem sie noch nicht zur Versionskontrolle hinzugefügt worden ist.

```
> nano 3BHIT.info
> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    3BHIT.info

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

10. Schließe alle `.info`-Dateien von der Versionierung aus und überprüfe danach den Status von `git`

Dies kann man mit einer `.gitignore`-Datei machen. `*.info` bedeutet, dass alle Dateien von der Versionskontrolle ausgeschlossen werden, welche die Endung `.info` besitzen.

```
> echo "*.info" >> .gitignore
> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

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

11. Füge alle Änderungen als neue Version zu deinem Repository hinzu und zeige die Log-Einträge deiner Versionen im Repository an.

```
> git commit -m "Add .gitignore"
[master befefae] Add .gitignore
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
```

```
commit befefae63e889ae2de15a7f05aa1cafe8c84f1ae (HEAD -> master)
Author: Andreas Suender <asuender@student.tgm.ac.at>
Date: Fri Nov 12 12:36:49 2021 +0100

    Add .gitignore

commit 4686569dc35d1a9c9f9e2e810f96573a4461e7de
Author: Andreas Suender <asuender@student.tgm.ac.at>
Date: Fri Nov 12 12:11:23 2021 +0100

    Add Suender.txt
(END)
```

12. Erstelle einen neuen leeren Unterordner mit dem Namen *Vorname* und überprüfe den Status von git.

Wie man hier schön sieht, werden leere Ordner von git nicht berücksichtigt.

```
> mkdir Andreas
> git status
On branch master
nothing to commit, working tree clean
```

13. Füge in den Unterordner eine Datei mit dem Namen *Vorname.txt* ein, die deinen Vornamen als Text enthält. Ändere in deinem Basis-Ordner die Datei *Nachname.txt* so dass die Ziffer den Wert 3 hat, speichere und überprüfe den Status von git

Nachdem ich eine Datei im *Andreas/* Ordner erstellt hatte, schien dieser auch schon rot in git status auf. Des Weiteren habe ich die *Suender.txt* wie vorgegeben verändert.

```

> cd Andreas
> nano Andreas.txt
> cd ..
> nano Suender.txt
> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Suender.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Andreas/

no changes added to commit (use "git add" and/or "git commit -a")

```

14. Füge alle Dateien mit * zum Staging-Bereich hinzu und überprüfe den Status von git.

```

> git add .
> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Andreas/Andreas.txt
        modified:   Suender.txt

```

15. Erstelle eine neue Version und zeige deinen Versionsverlauf (Log-Einträge) an.

```

> git commit -m "Create/Modifiy Andreas.txt and Suender.txt"
[master 5be2367] Create/Modifiy Andreas.txt and Suender.txt
2 files changed, 2 insertions(+), 1 deletion(-)
create mode 100644 Andreas/Andreas.txt

```

```
commit 5be23675044f1718c3114c5e96aa89812716520b (HEAD -> master)
Author: Andreas Suender <asuender@student.tgm.ac.at>
Date:   Fri Nov 12 12:46:13 2021 +0100
```

Create/Modifiy Andreas.txt and Suender.txt

```
commit befefae63e889ae2de15a7f05aa1cafe8c84f1ae
Author: Andreas Suender <asuender@student.tgm.ac.at>
Date:   Fri Nov 12 12:36:49 2021 +0100
```

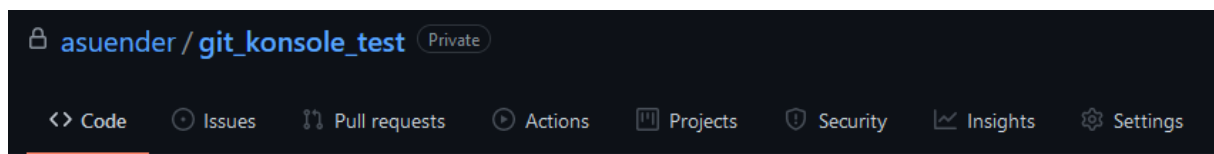
Add .gitignore

```
commit 4686569dc35d1a9c9f9e2e810f96573a4461e7de
Author: Andreas Suender <asuender@student.tgm.ac.at>
Date:   Fri Nov 12 12:11:23 2021 +0100
```

Add Suender.txt

(END)

16. Erstelle auf Github ein neues leeres Repository unter eurem Github Account



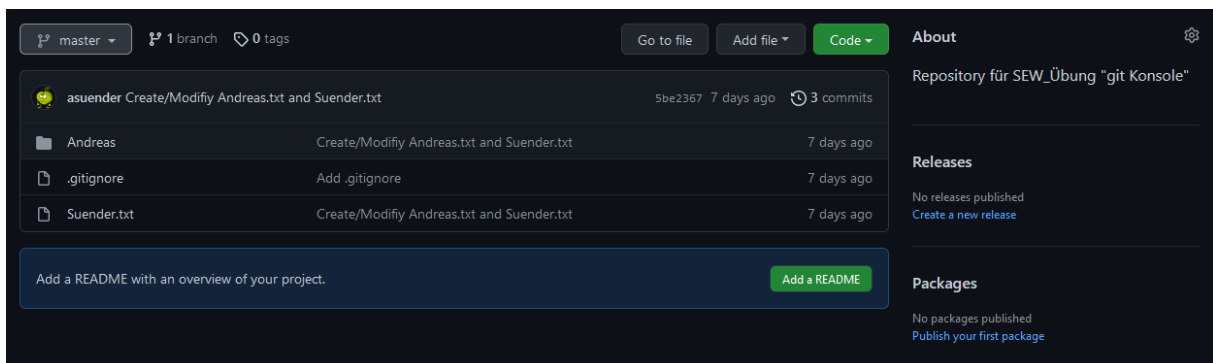
17. Füge dieses Github Repository bei deinem lokalen Repository als Remote-Repository unter dem origin hinzu

Mit dem Befehl `git remote add <remote> <url>` kann man ein Remote-Repository mit dem Namen `<remote>` hinzufügen. In `git remote -v` scheint dieses dann sowohl fürs Fetchen als auch Pushen auf.

```
> git remote add origin https://github.com/asuender/git_konsole_test.git
> git remote -v
origin https://github.com/asuender/git_konsole_test.git (fetch)
origin https://github.com/asuender/git_konsole_test.git (push)
```


18. Übertrage die Versionen von deinem lokalen Repository auf dein Github-Repository und überprüfe dann dort, ob deine Dateien vorhanden sind und ob du auch alle Versionen siehst.

```
> git push origin master
Username for 'https://github.com': asuender
Password for 'https://asuender@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (11/11), 879 bytes | 125.00 KiB/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To https://github.com/asuender/git_konsole_test.git
* [new branch]      master -> master
```



The screenshot shows the GitHub interface for a repository named "Repository für SEW_Übung "git Konsole"". The repository is owned by "asuender" and has 1 branch and 0 tags. The commit history shows a single commit by "asuender" titled "Create/Modify Andreas.txt and Suender.txt" from 7 days ago. The file list includes "Andreas", ".gitignore", and "Suender.txt". The right sidebar shows the "About" section with the repository name, "Releases" section with no published releases, and "Packages" section with no published packages. A button "Add a README" is visible at the bottom left.

Es ist alles da 😊!