# Ein erstes Repo

## Initialisieren

Wir wollen zunächst eine kleine statische Webseite entwickeln und unter Versionskontrolle stellen.

```
$ mkdir webseite
$ cd webseite
$ git init
Leeres Git-Repository in /home/frank/Downloads/webseite/.git/ initialisiert
```

Nun steht das Verzeichnis `webseite` unter Versionskontrolle. Das (lokale) Git-Repository befindet sich im Unterverzeichnis `.git`:

```
$ ls -la
insgesamt 132
drwxr-xr-x  3 frank frank   4096 24. Okt 13:32 .
drwxr-xr-x 21 frank frank 122880 24. Okt 13:32 ..
drwxr-xr-x  7 frank frank   4096 24. Okt 13:32 .git
```

## Grundkonfiguration

Bevor man sinnvoll mit Git arbeiten kann, sollte man zunächst seinen Namen und seine Mailadresse korrekt einstellen:

```
git config user.email "meine@mail.adresse.hier"
git config user.name "John Doe"
```

Diese Befehle speichern die Einstellungen nur für das Repository, in dem wir gerade arbeiten, wenn man die Einstellungen für alle Repos auf einem Rechner vornehmen möchte, muss man das Flag `--global` ergänzen.

Gespeichert werden die für ein Repository geltenden lokalen Einstellungen in der Datei `.git/config`:

```
[frank@rita webseite]$ git config  user.name "John Doe"
[frank@rita webseite]$ git config  user.email "john.doe@nirgends.nix"
[frank@rita webseite]$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[user]
    name = John Doe
    email = john.doe@nirgends.nix
```

# Repository Status anzeigen lassen

Mit dem Befehl `git status` kann man sich den aktuellen Status des Repos anzeigen lassen:

```
[frank@rita webseite]$ git status
Auf Branch main

Noch keine Commits

nichts zu committen (erstellen/kopieren Sie Dateien und benutzen
Sie "git add" zum Versionieren)
```

# Ein erster Commit

Den aktuellen Status eines Repositorys kann man sich durch den Befehl `git status` anzeigen lassen:

```
$ git status
Auf Branch master

Noch keine Commits

nichts zu committen (erstellen/kopieren Sie Dateien und benutzen
Sie "git add" zum Versionieren)
```

Lege nun eine `index.html` Datei ein sowie zwei Verzeichnisse - `css` und `img`:

```
sbel@r107-ws15:~/git$ touch index.html
sbel@r107-ws15:~/git$ mkdir css
sbel@r107-ws15:~/git$ mkdir img
sbel@r107-ws15:~/git$ ls
css  img  index.html
```

- Neue Dateien befinden sich zunächst im Arbeitsverzeichnis und werden von git ignoriert. Teste das mit `git status`.
- Mit dem Befehl `git add` wird eine Datei in gits „Staging Area“ verschoben - das kann man sich vorstellen wie ein Einkaufswagen, in dem neue Dateien und Änderungen gesammelt werden, bis man zu einem Punkt kommt, den man sich „merken“ möchte. Dann macht man einen „Commit“. Füge die Datei `index.html` deiner Staging Area hinzu und kontrolliere das Ergebnis mit `git status`
- Führe den Befehl `git commit` aus, gib eine Commit-Message an.
- Überprüfe den Zustand von Arbeitsverzeichnis und Staging Area mit `git status`. Schau dir die Liste deiner Commits mit `git log` an.

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
**https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:git:erstes_repo:start?rev=1619615011**

Last update: **28.04.2021 15:03**