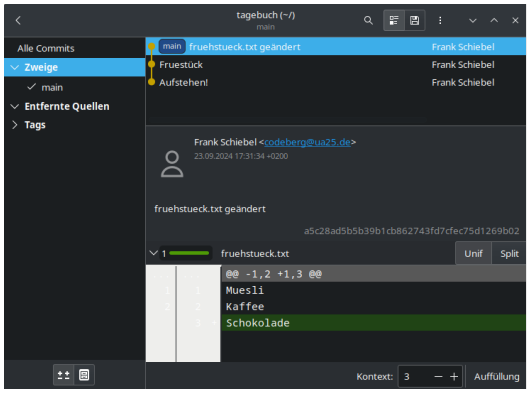
Szenario: Zurück in der Zeit und nochmal versuchen

Ein Szenario das man gelegentlich hat, ist, dass man in der Zeit zurückgehen möchte, dann aber einen Fehler macht, den man gerne korrigieren möchte.

Noch einmal das Tagebuch

Wir starten mit dem folgenden Zustand - im Commit a5c28ad wurde dem Frühstück Schokolade hinzugefügt.




```
frank@pike:~/tagebuch$ git lg
* a5c28ad - (HEAD -> main) fruehstueck.txt geändert (vor 22 Stunden)
* 2c70b75 - Fruestück (vor 11 Monaten)
* 9ee8f8b - Aufstehen! (vor 11 Monaten)
```

Wir wollen nun zu dem Zeitpunkt zurückgehen, zu dem das Frühstück nur aus zwei Speisen bestand, um dort anstelle der Schokolade eine Banane zu verspeisen.

Dazu gibt es mehrere Möglichkeiten, die verschiedene Vor- und Nachteile haben.

Einfach zurück und neu starten - bisherige Änderungen gehen verloren

> Dieses Vorgehen ist eine der (wenigen) Möglichkeiten in git tatsächlich **Informationen unwiederbringlich zu verlieren**. Man sollte sich also wirklich sicher sein, dass man die Änderungen, die man nach dem Zeitpunkt, zu dem man zurückkehren möchte nicht mehr benötigt.

Zur Verdeutlichung dieses Situation habe ich jetzt noch eine weitere neue Datei angelegt - diese heißt wieder `mittagessen.txt`, steht aber noch nicht unter Versionsverwaltung.

```
frank@pike:~/tagebuch$ git status
Auf Branch main
Unversionierte Dateien:
  (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit
vorzumerken)
```

```
    mittagessen.txt

nichts zum Commit vorgemerkt, aber es gibt unversionierte Dateien
(benutzen Sie "git add" zum Versionieren)
frank@pike:~/tagebuch$ git lg
* a5c28ad - (HEAD -> main) fruehstueck.txt geändert (vor 23 Stunden)
* 2c70b75 - Fruestück (vor 11 Monaten)
* 9ee8f8b - Aufstehen! (vor 11 Monaten)
```

Mit dem Befehl `git reset –hard 2c70b75` bringt man alle Dateien im Verzeichnis, die unter Versionskontrolle stehen auf den Stand, die diese zum Zeitpunkt des Commits `2c70b75` hatten. Alle späteren Änderungen und die spätere Versionsgeschichte gehen unwiderruflich verloren! Dateien, die *nicht* unter Versionskontrolle stehen, werden nicht beeinflusst.

Das sieht dann so aus:

```
frank@pike:~/tagebuch$ git reset --hard 2c70b75
HEAD ist jetzt bei 2c70b75 Fruestück
frank@pike:~/tagebuch$ ls
aufstehen.txt  fruehstueck.txt  mittagessen.txt
frank@pike:~/tagebuch$ git lg
* 2c70b75 - (HEAD -> main) Fruestück (vor 11 Monaten)
* 9ee8f8b - Aufstehen! (vor 11 Monaten)
frank@pike:~/tagebuch$ git lg --all
* 2c70b75 - (HEAD -> main) Fruestück (vor 11 Monaten)
* 9ee8f8b - Aufstehen! (vor 11 Monaten)
```

- Die Dateien `aufstehen.txt` und `fruehstueck.txt` sehen jetzt so aus, wie sie zum Zeitpunkt es Commits aussahen
- Die Datei `mittagessen.txt` ist unverändert, weil sie nicht unter Versionskontrolle stand
- `git log` zeigt auch mit der Option `--all` die Commits, die nach `2c70b75` gemacht wurden nicht mehr an - diese Daten sind verloren.

> Jetzt kann man sein Tagebuch ausgehend von `2c70b75` weiter führen, hat das alternative Frühstück mit Schokolade aber verloren.

Zurück und mit einem weiteren Branch weiterarbeiten - die bisherigen Änderung bleiben erhalten

Schritt 1 - Checkout des letzten Commits, der beibehalten werden soll