

Nicht immer möchte man alle Änderungen in einen Commit aufnehmen.

Hier die Anleitung

# Zeilenweise Committen

```
$ git add -p <filename>
```

- **y**  
diesen Patch-Block zum Commit vormerken
- **n**  
diesen Patch-Block nicht zum Commit vormerken
- **q**  
Beenden; diesen oder alle verbleibenden Patch-Blöcke nicht zum Commit vormerken
- **a**  
diesen und alle weiteren Patch-Blöcke dieser Datei zum Commit vormerken
- **d**  
diesen oder alle weiteren Patch-Blöcke in dieser Datei nicht zum Commit vormerken
- **g**  
Patch-Block zum Hinspringen auswählen
- **/**  
nach Patch-Block suchen der gegebenem regulärem Ausdruck entspricht
- **j**  
diesen Patch-Block unbestimmt lassen, nächsten unbestimmten Patch-Block anzeigen
- **J**  
diesen Patch-Block unbestimmt lassen, nächsten Patch-Block anzeigen
- **k**  
diesen Patch-Block unbestimmt lassen, vorherigen unbestimmten Patch-Block anzeigen
- **K**  
diesen Patch-Block unbestimmt lassen, vorherigen Patch-Block anzeigen
- **s**  
aktuellen Patch-Block in kleinere Patch-Blöcke aufteilen
- **e**  
aktuellen Patch-Block manuell editieren
- **?**  
Hilfe anzeigen

Alternativ können die zu übernehmenden Änderungen auch im Editor Ihrer Wahl ausgewählt werden:

```
$ git add -e <filename>
```

Nach Aufruf des Befehls öffnet sich der Standard-Editor. Um eine Zeile hinzuzufügen/zu entfernen muss man das + oder - mit einem # ersetzen.

Schließlich können die ausgewählten Abschnitte mit

```
$ git commit
```

übertragen werden.

## Quelle:

- <https://www.veit-schiele.de/dienstleistungen/technische-dokumentation/git/git-commit-einzelner-abschnitte>

From:  
<https://wiki.da-checka.de/> - PSwiki



Permanent link:  
[https://wiki.da-checka.de/doku.php/wiki/programme/git/commit\\_zeilenweise](https://wiki.da-checka.de/doku.php/wiki/programme/git/commit_zeilenweise)

Last update: **2022/02/16 14:15**