Änderungen untersuchen

Änderungen zwischen Commits ansehen

Wir arbeiten weiter in unserem Beispielrepository für das Tagebuch. Zunächst stellen wir sicher, dass wir auf den aktuellen `main`-Branch ausgecheckt haben:

```
frank@pike:~/tagebuch$ git checkout main
Bereits auf 'main'
frank@pike:~/tagebuch$ git lg
* 022bdbc - (HEAD -> main) Mittagessen hinzugefügt (vor 20 Stunden)
* a5c28ad - fruehstueck.txt geändert (vor 20 Stunden)
* 2c70b75 - Fruestück (vor 11 Monaten)
* 9ee8f8b - Aufstehen! (vor 11 Monaten)
```

Vom HEAD zurück blicken

Wenn man vom HEAD zurückblicken möchte, kann man folgenden Befehl verwenden `git diff HEAD~1`. Das bedeutet: "Zeige mir alle Unterschiede im Verzeichnis zwischen dem Commit, auf den HEAD gerade zeigt und dem vorigen Commit" - die Ausgabe ist zunächst etwas gewöhnungsbedürftig:

```
frank@pike:~/tagebuch$ git diff HEAD~1
diff --git a/mittagessen.txt b/mittagessen.txt
new file mode 100644
index 0000000..4f90f20
--- /dev/null
+++ b/mittagessen.txt
@@ -0,0 +1 @@
+Suppe!
```

Die Ausgabe sagt uns:

- Es wurde eine neue Datei angelegt - `mittagessen.txt`
- Dort wurde eine Zeile eingefügt: `Suppe!`

Man kann sehr einfach auch zwei Commits in die Vergangenheit blicken: `git diff HEAD~2`:

```
frank@pike:~/tagebuch$ git diff HEAD~2
diff --git a/fruehstueck.txt b/fruehstueck.txt
index c8b8882..347ae7f 100644
--- a/fruehstueck.txt
+++ b/fruehstueck.txt
@@ -1,2 +1,3 @@
 Muesli
 Kaffee
+Schokolade
diff --git a/mittagessen.txt b/mittagessen.txt
```

```
new file mode 100644
index 0000000..4f90f20
--- /dev/null
+++ b/mittagessen.txt
@@ -0,0 +1 @@
+Suppe!
```

Die Ausgabe sagt uns, dass in allen Commits bis zum aktuellen HEAD die folgenden Änderungen im Repo stattgefunden haben:

- Es wurde eine neue Datei angelegt - `mittagessen.txt`
- Dort wurde eine Zeile eingefügt: `Suppe!`
- In der zuvor bereits vorhandenen Datei `fruehstueck.txt` wurde nach den beiden schon vorhandenen Zeilen eine weitere Zeile `Schokolade` eingefügt.

(A1)

- Untersuche die Unterschiede in deinem Repo zwischen dem HEAD auf main und einigen vorigen Commits
- Erstelle auf dem main Branch einen weiteren Commit, bei dem du in einer deiner Dateien eine Zeile entfernst. Untersuche, wie die Ausgabe von `git diff` jetzt aussieht - woran erkennt man, dass die Zeile entfernt wurde?
- Erstelle einen Commit, bei dem du eine Datei entfernst (`git rm <Dateiname>`, dann einen Commit erstellen), untersuche, wie die Ausgabe von `git diff` jetzt aussieht

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:git:diff:start?rev=1727178466

Last update: **24.09.2024 11:47**