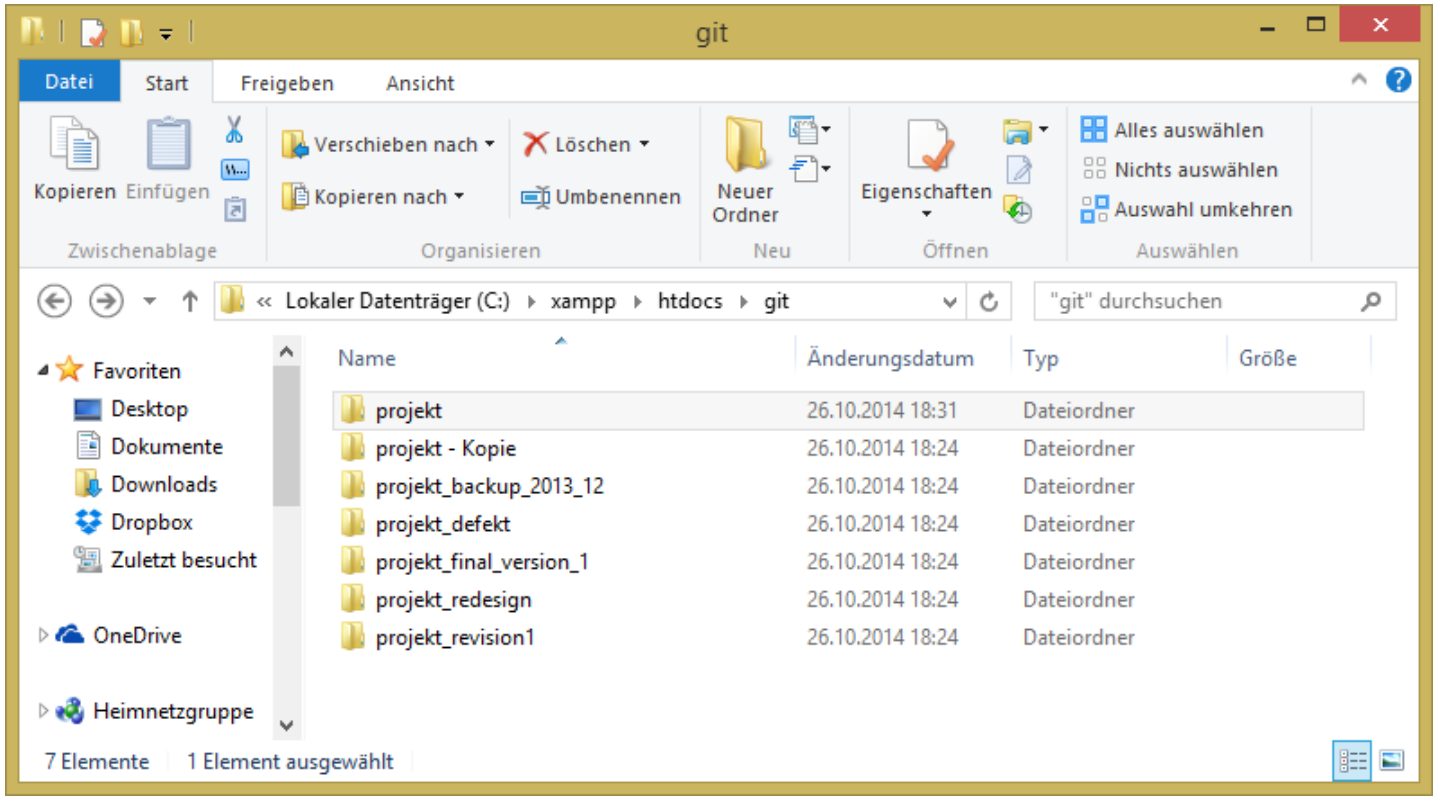
Versionsverwaltung mit Git

4. WP Meetup Berlin Workshop

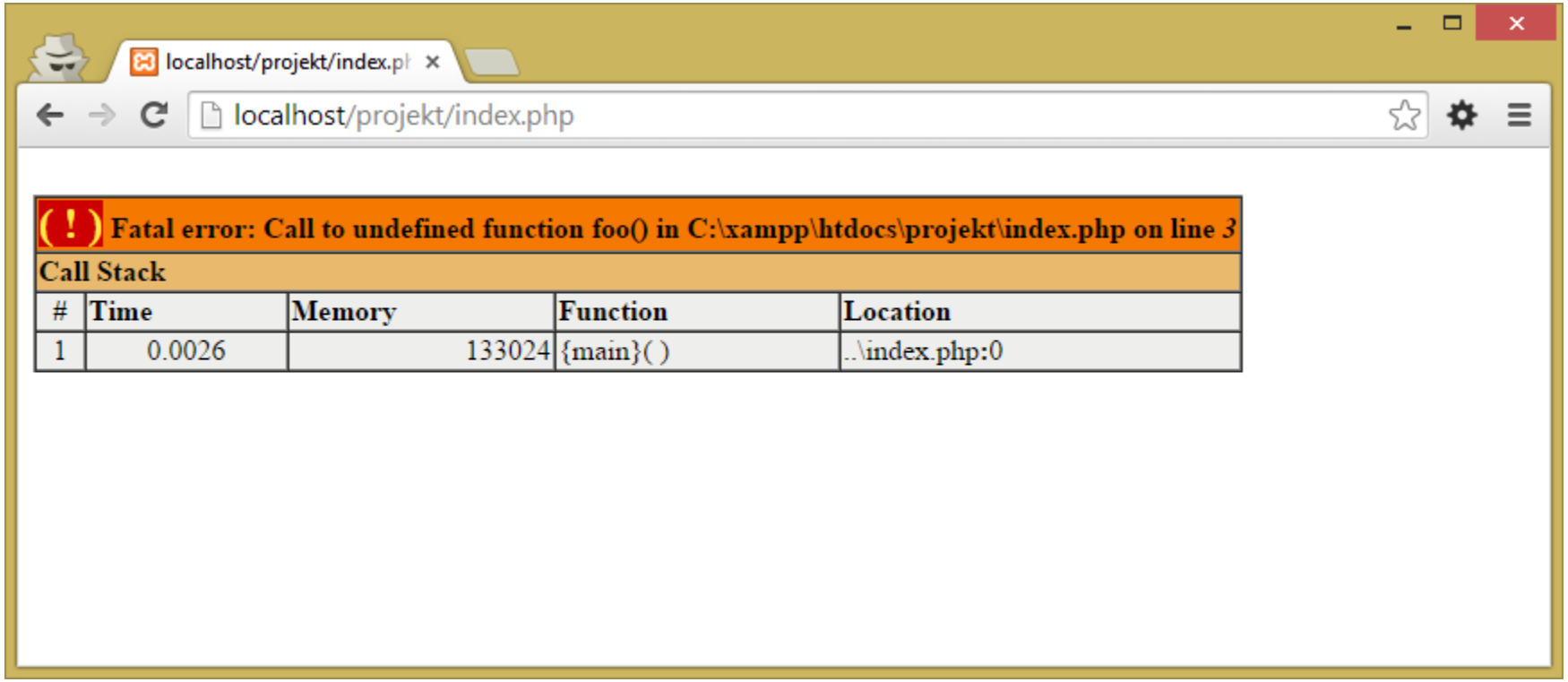
Was ist Git eigentlich?

- Git ist ein verteiles (dezentrales) System zur Versionsverwaltung von Dateien.
- Es ist eine moderne Alternative zu zentralen Versionsverwaltungssystemen wie beispielsweise Subversion (SVN)
- Entwickelt wurde es von Linus Torvalds , dem Entwickler des Linux-Kernels.
- Git heißt übersetzt „Blödmann/Schwachkopf“ ;)

Wieso sollte ich Git einsetzen?



Wieso sollte ich Git einsetzen?



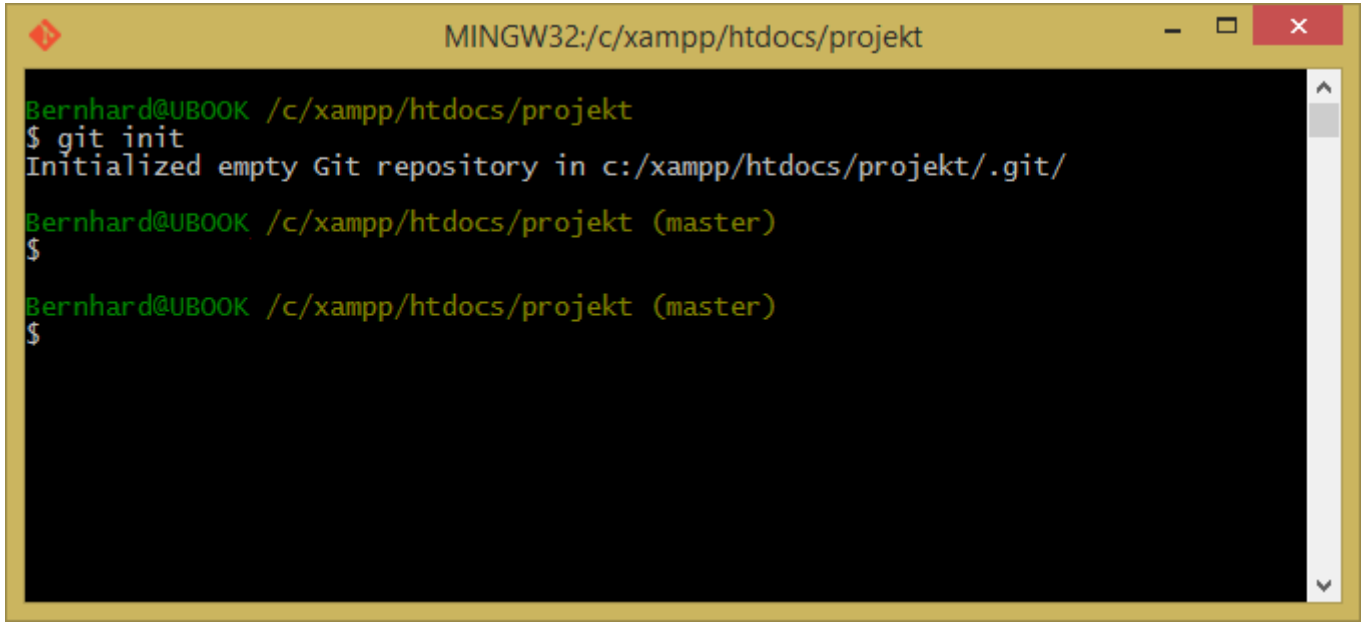
Die Git Basics

- Ein Repository erstellen
- Dateien zu einem Repository hinzufügen
- Änderungen an Dateien anzeigen
- Änderungen an Dateien sichern
- Änderungen rückgängig machen
- Arbeiten mit Entwicklungszweigen
- Projekt mit anderen teilen

Ein Repository erstellen

- Ein Repository aus einem (neuen) Projekt heraus erstellen

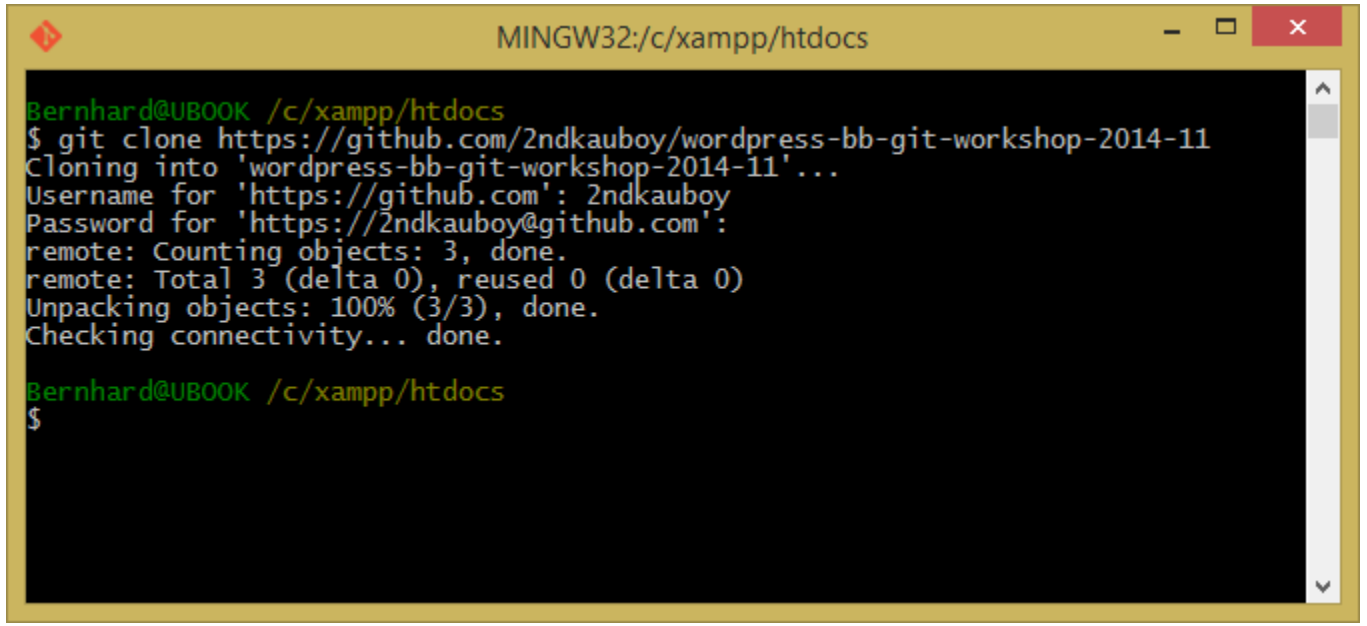
```
git init
```



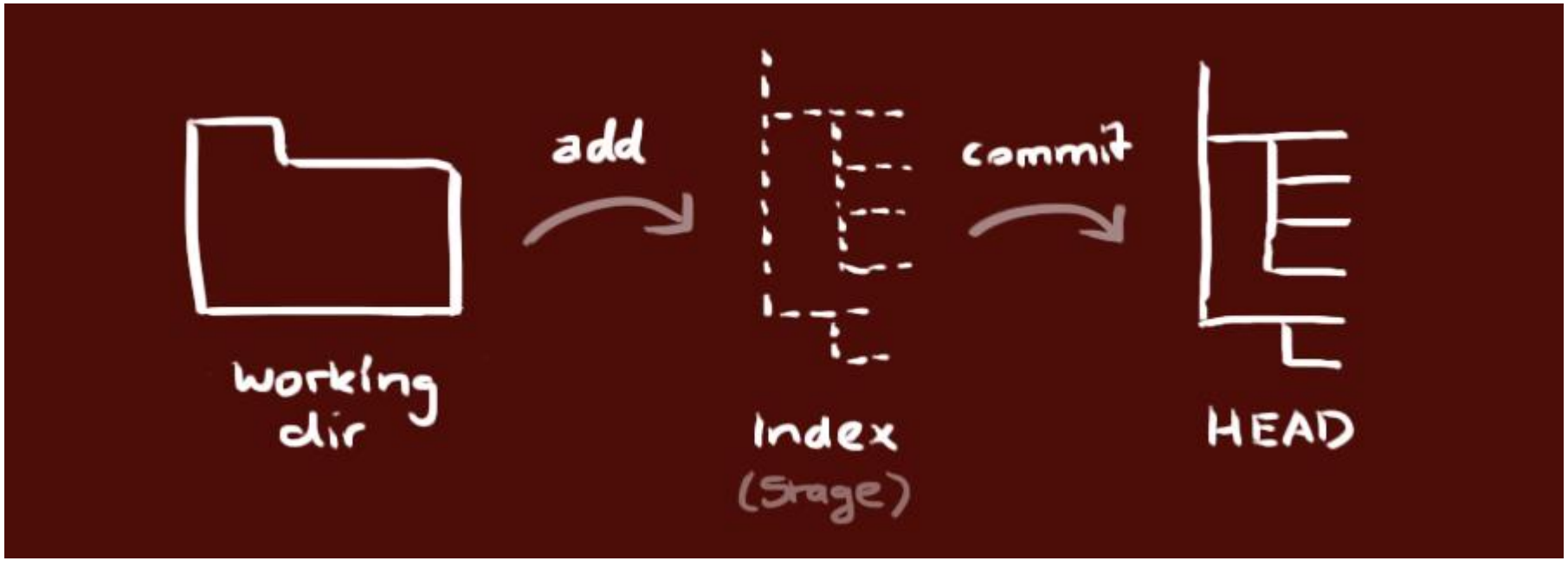
Ein Repository erstellen

- An einem bestehendes Projekt mitarbeiten – es „klonen“

```
git clone https://github.com/2ndkauboy/wordpress-bb-git-workshop-2014-11
```



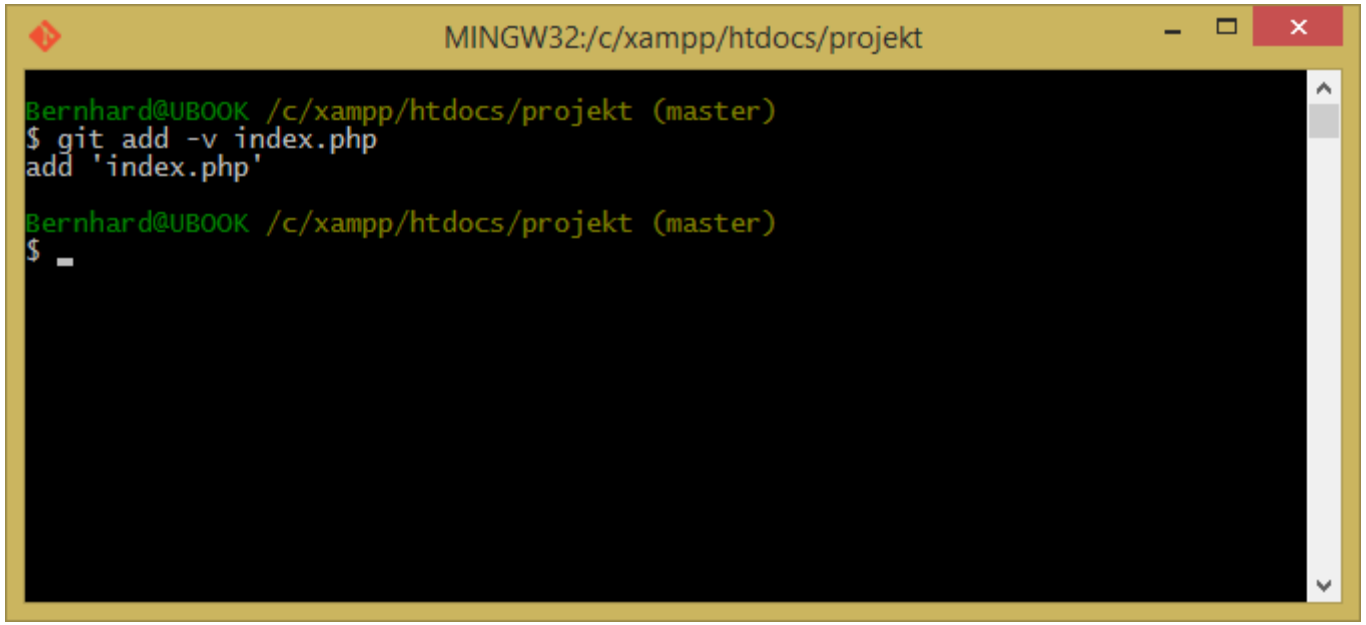
Dateien zu einem Repository hinzufügen



Dateien zu einem Repository hinzufügen

- Datei auf den „Index“ legen

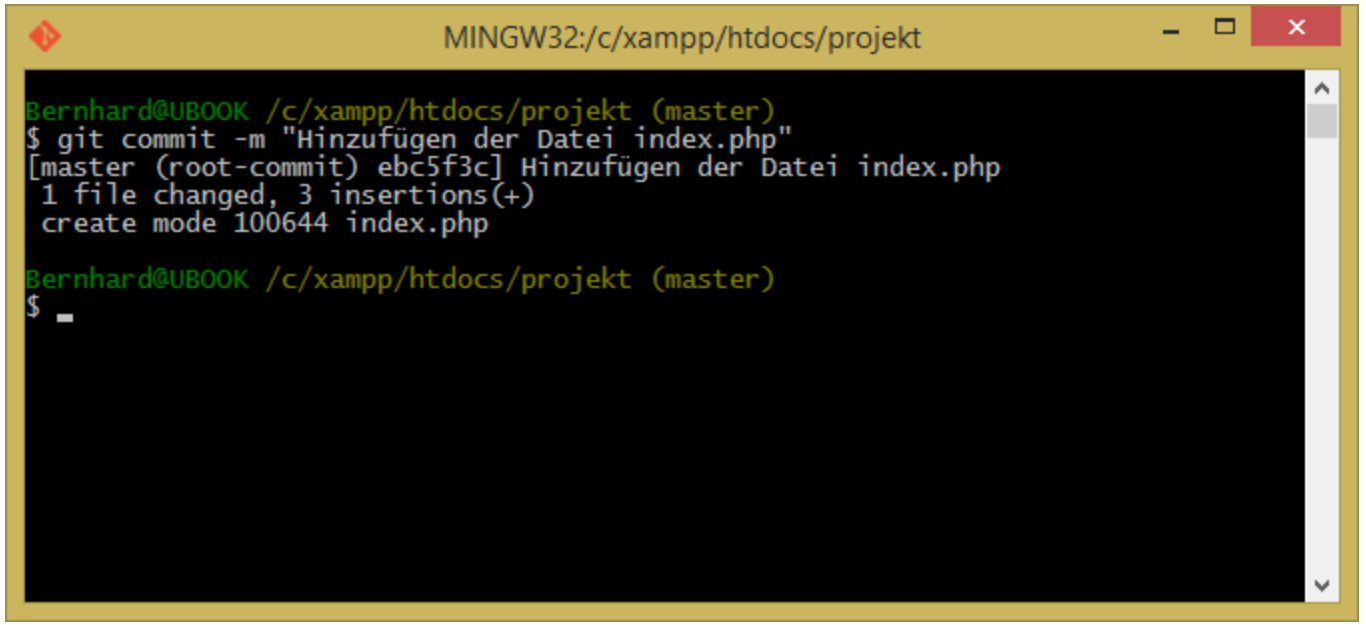
```
git add index.php
```



Dateien zu einem Repository hinzufügen

- Datei zum Respository hinzufügen

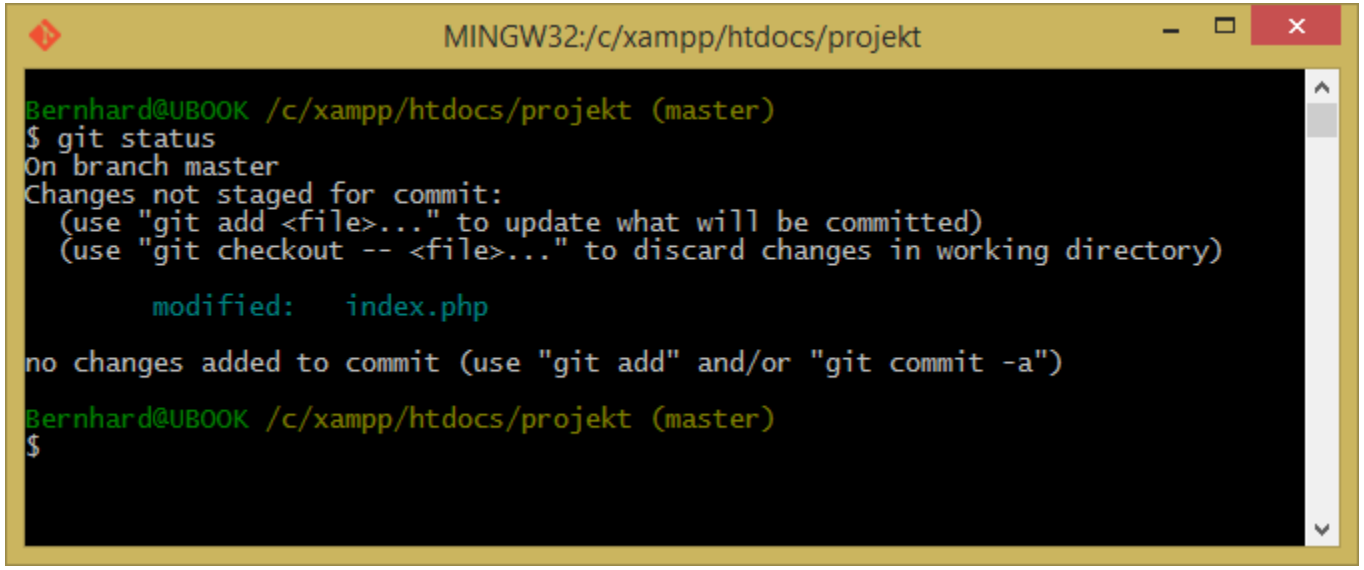
```
git commit -m "Hinzufügen der Datei index.php"
```



Änderungen an Dateien anzeigen

- Was hat sich seit dem letzten Commit geändert?

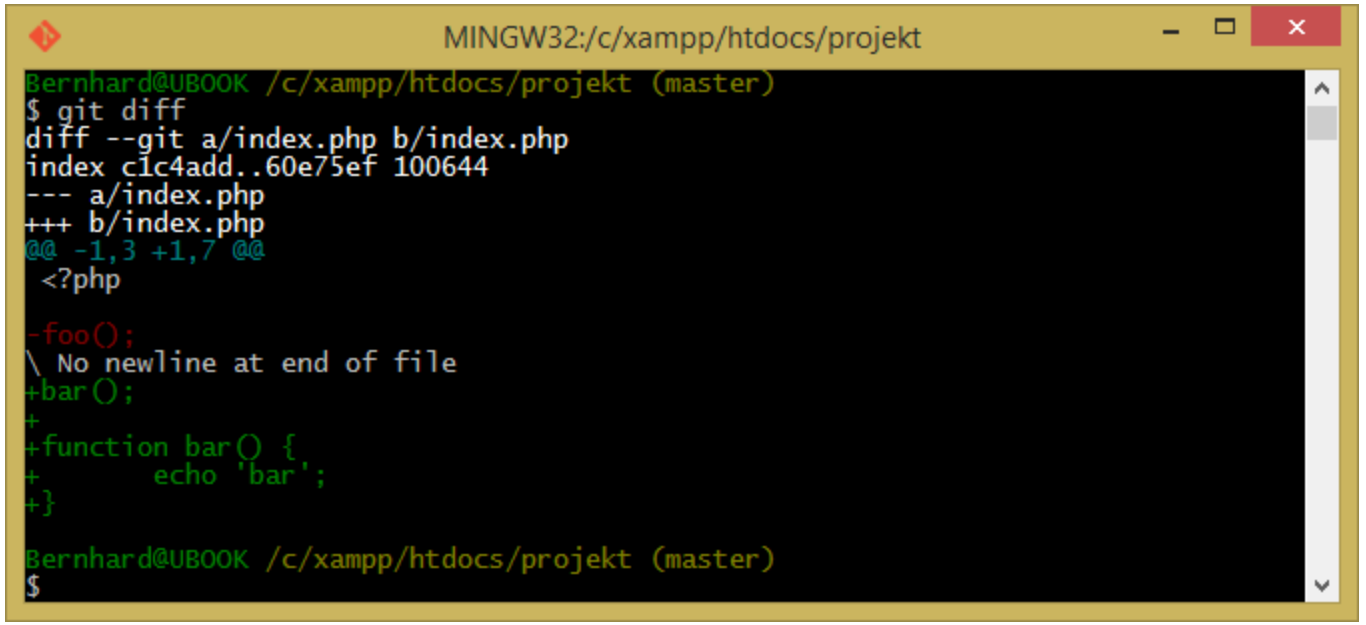
```
git status
```



Änderungen an Dateien anzeigen

- Was hat sich seit dem letzten Commit geändert?

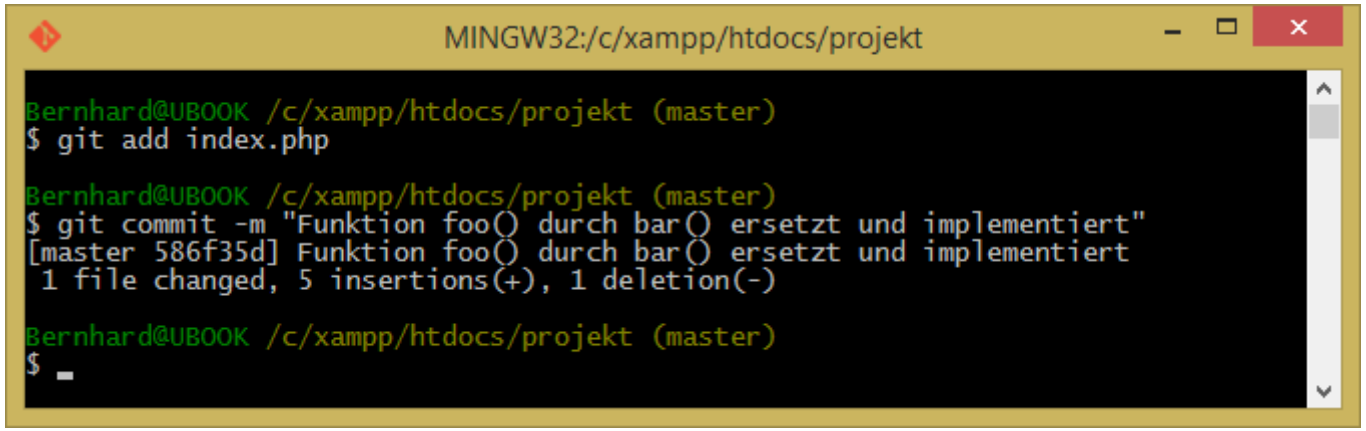
```
git diff
```



Änderungen an Dateien sichern

- Dateiänderungen vormerken und commiten

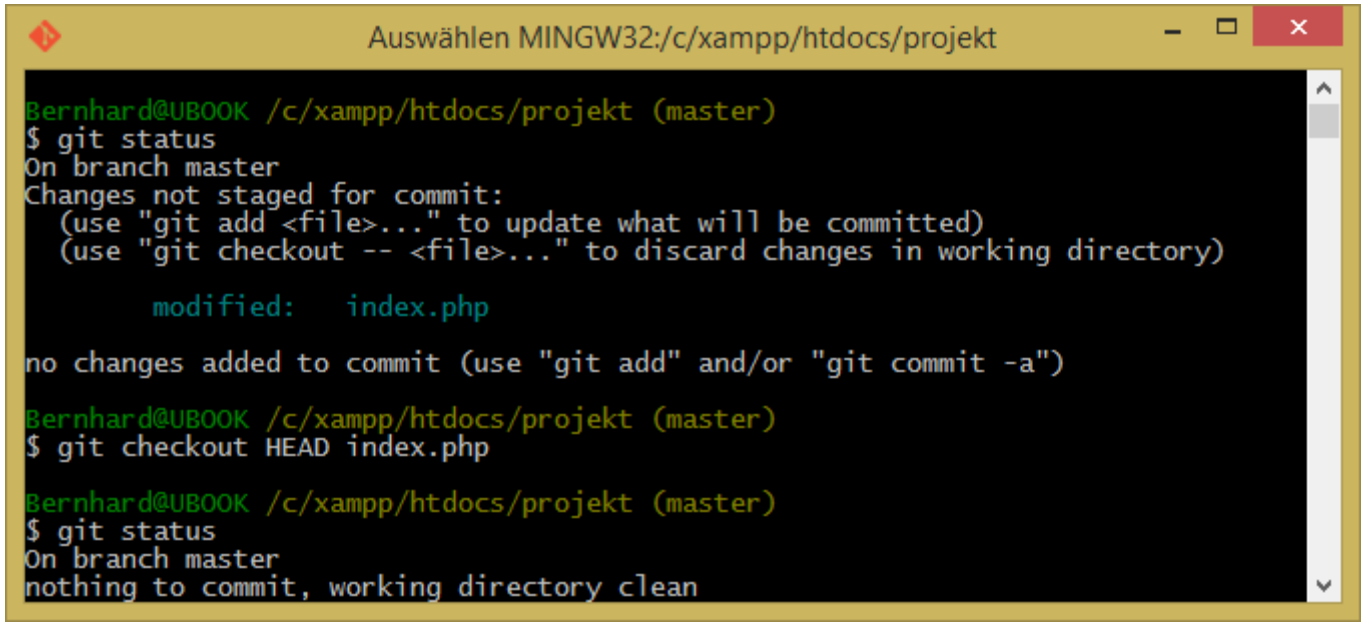
```
git add index.php
git commit -m "Funktion foo() …"
```



Änderungen rückgängig machen

- Änderungen an einer Datei seit dem letzten Commit verwerfen

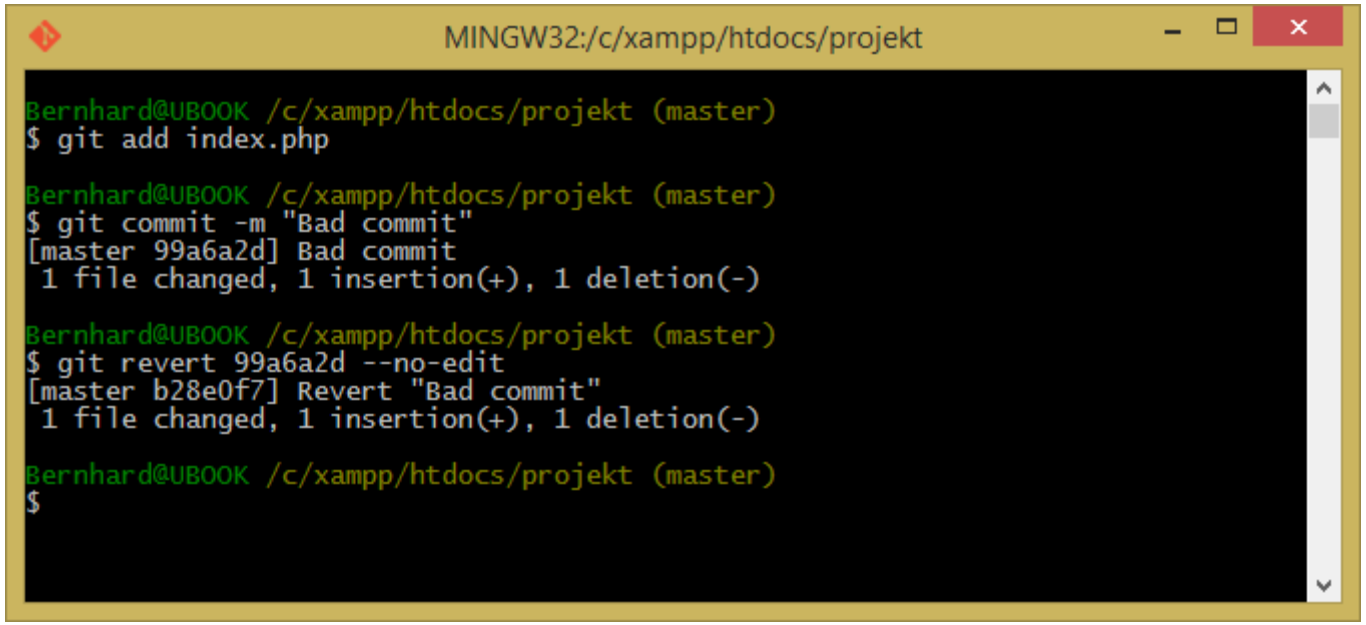
```
git checkout HEAD index.php
```



Änderungen rückgängig machen

- Alle Änderungen eines Commits rückgängig machen

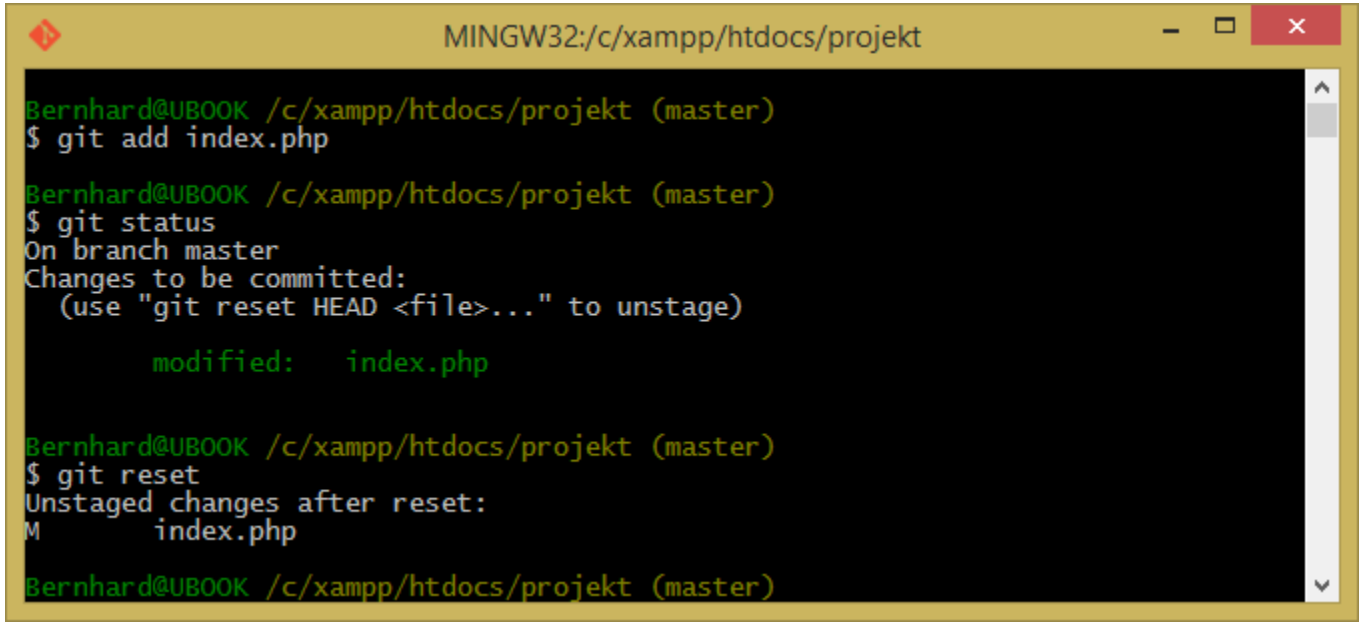
```
git revert <commit>
```



Änderungen rückgängig machen

- Datei vom Index nehmen (ein „git add“ rückgängig machen)

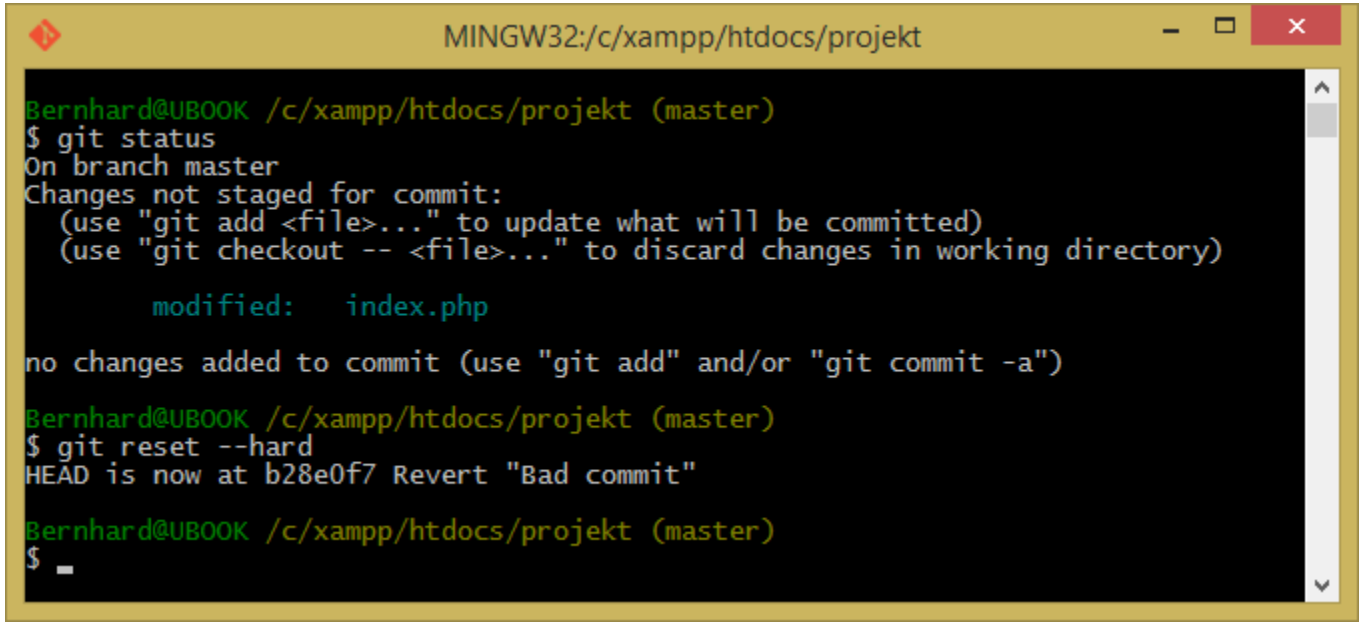
```
git reset [datei]
```



Änderungen rückgängig machen

- Alle lokalen Änderungen verwerfen und Index zurücksetzen – GEFÄHRLICH!

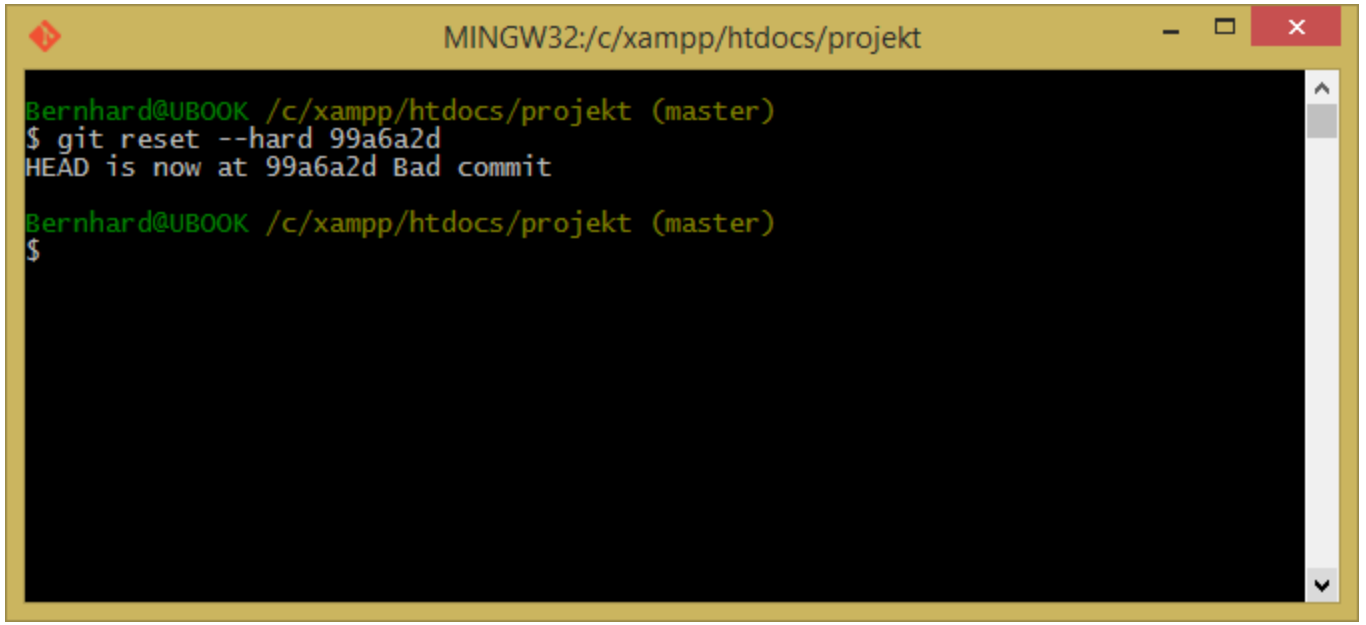
```
git reset --hard
```



Änderungen rückgängig machen

- Einen Commit vom Verlauf entfernen – GEFÄHRLICH!

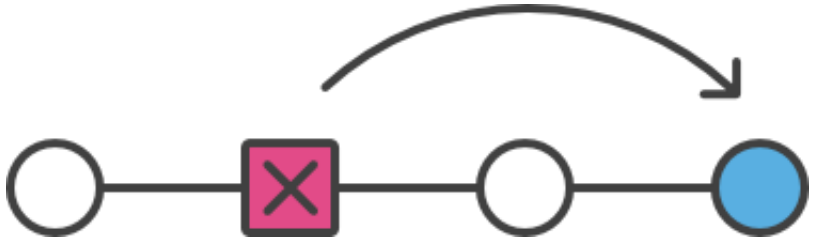
```
git reset –hard <commit>
```



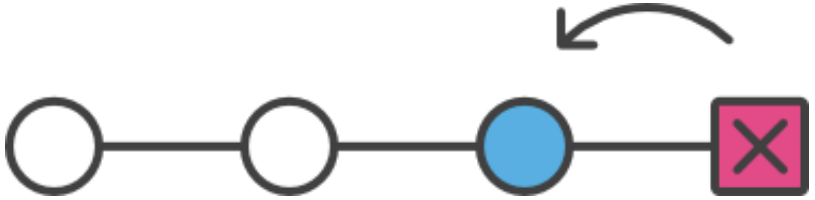
Änderungen rückgängig machen

- Der Unterschied zwischen „revert“ und „reset --hard“

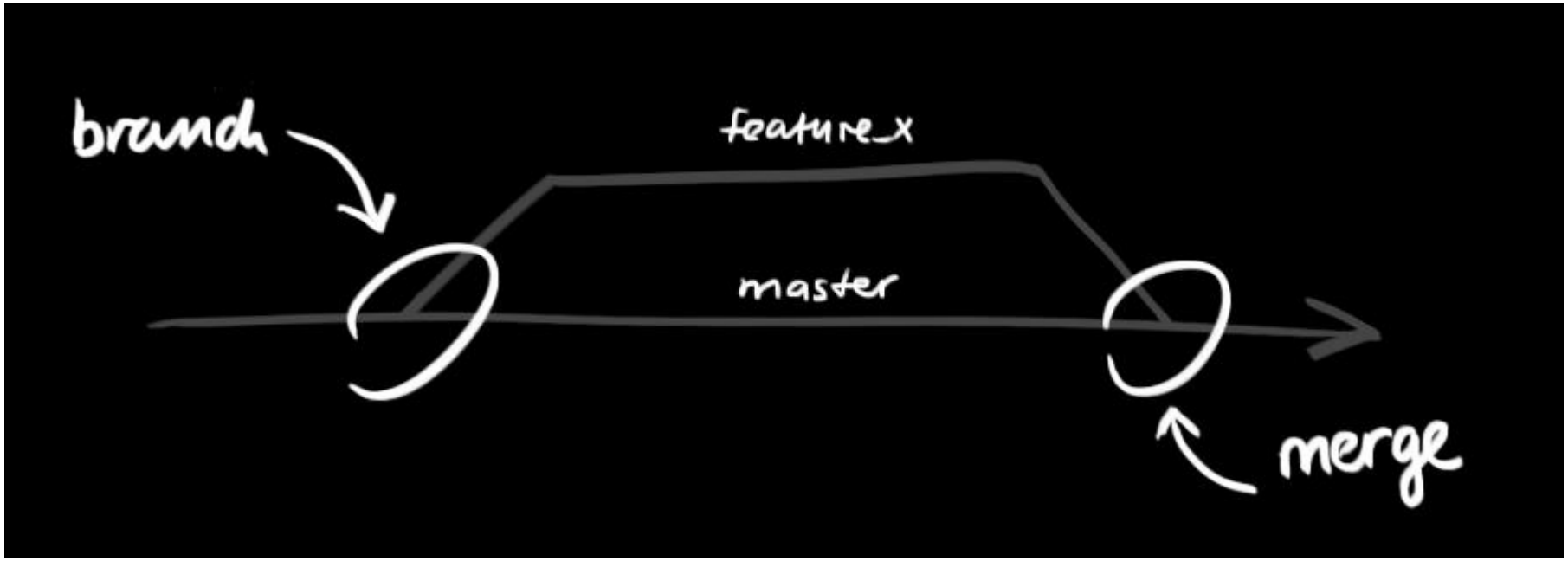
Revert



Reset



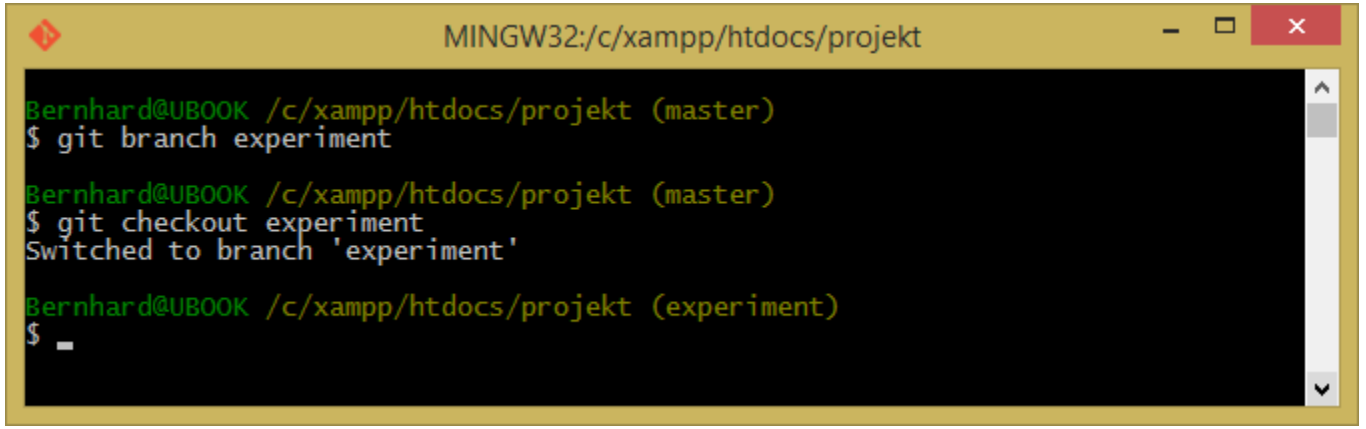
Arbeiten mit Entwicklungszweigen



Arbeiten mit Entwicklungszweigen

- Neuen Entwicklungszweig („Branch“) erstellen und dorthin wechseln

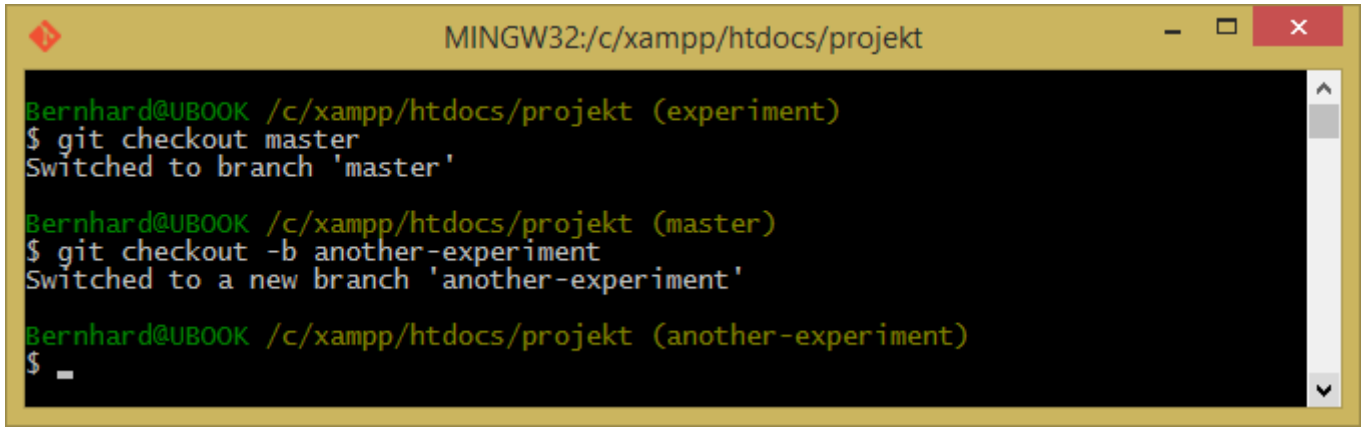
```
git branch <branch>
git checkout <branch>
```



Arbeiten mit Entwicklungszweigen

- Einen neuen Branch erstellen und direkt auschecken

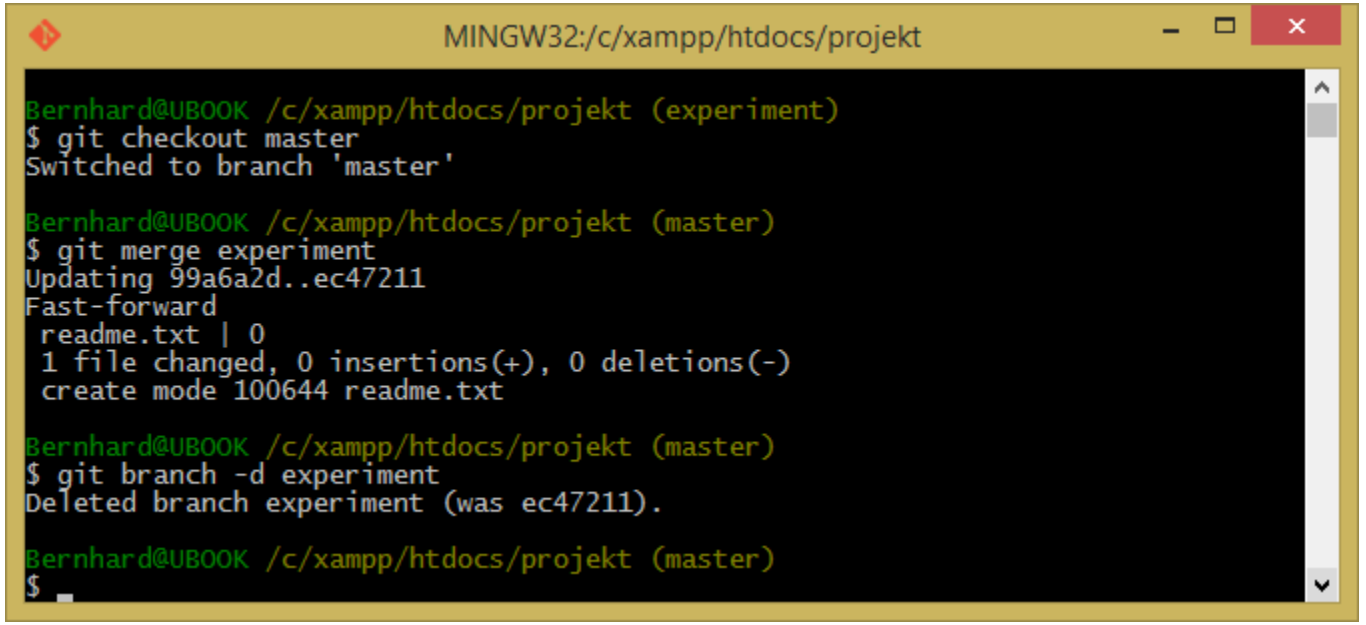
```
git checkout –b <branch>
```



Arbeiten mit Entwicklungszweigen

- Änderungen eines Branches in den Hauptbranch zurückführen („merge“)

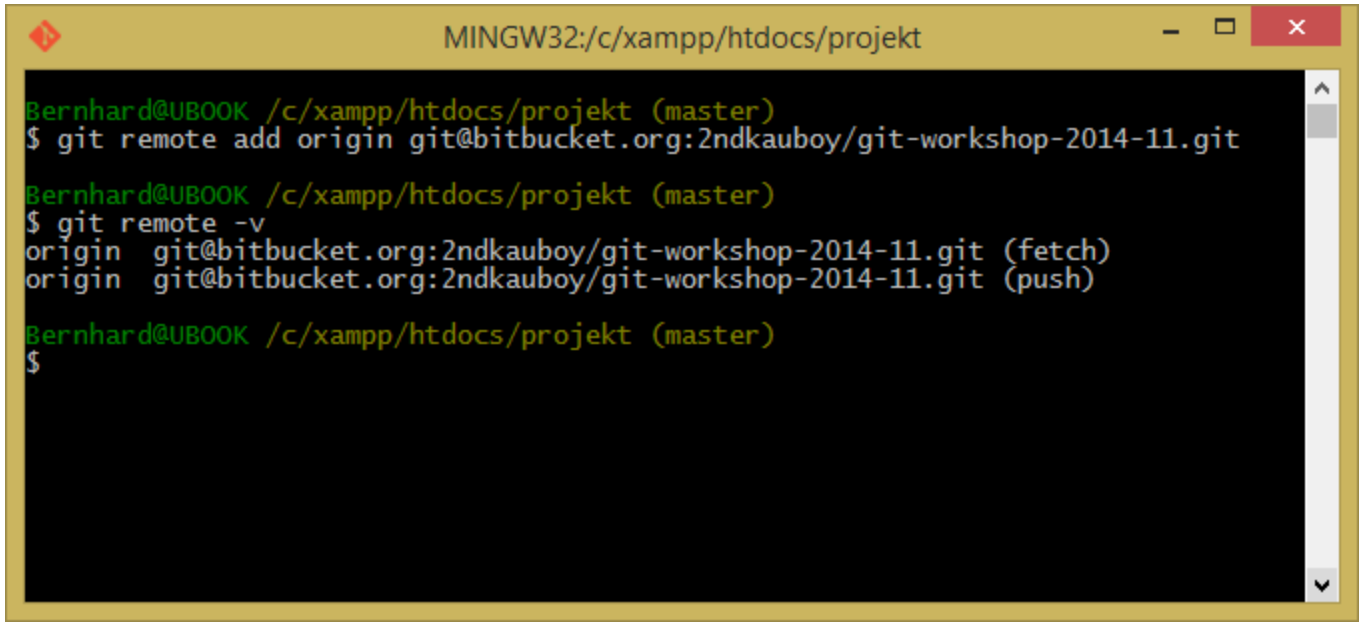
```
git merge <branch>
```



Projekt mit anderen teilen

- Projekt mit einem Remote-Repository verknüpfen

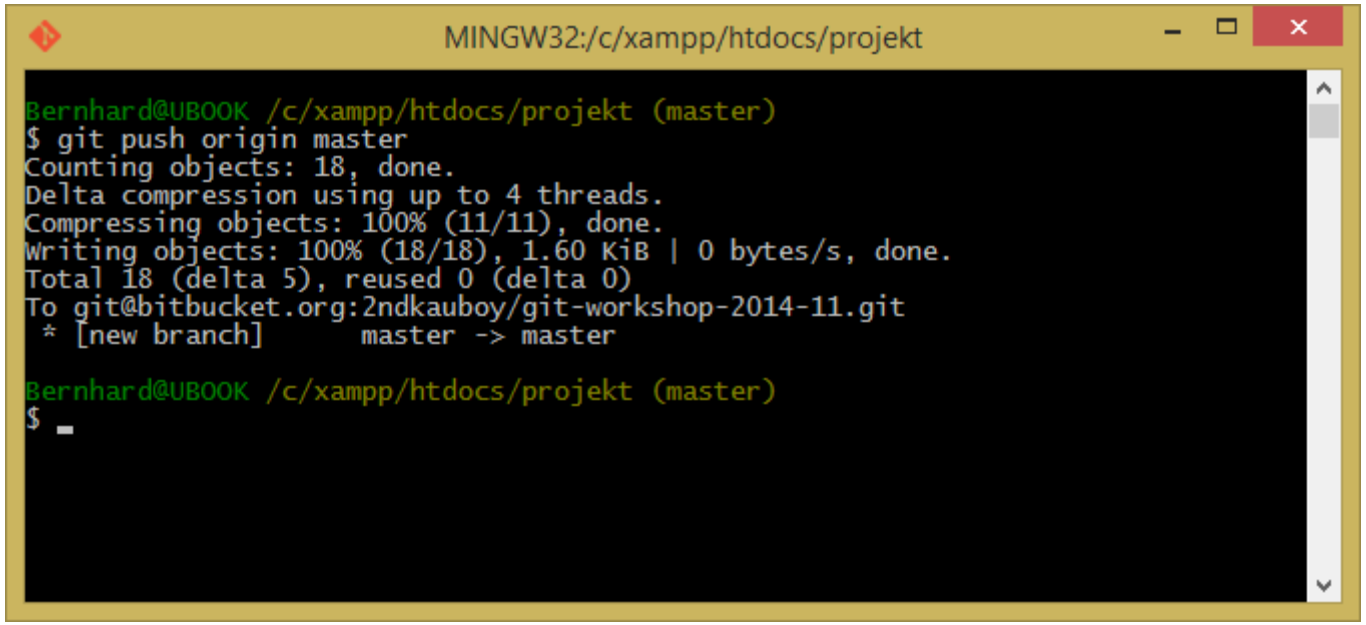
```
git remote add <name> <url>
```



Projekt mit anderen teilen

- Commits an das Remote-Repository schicken

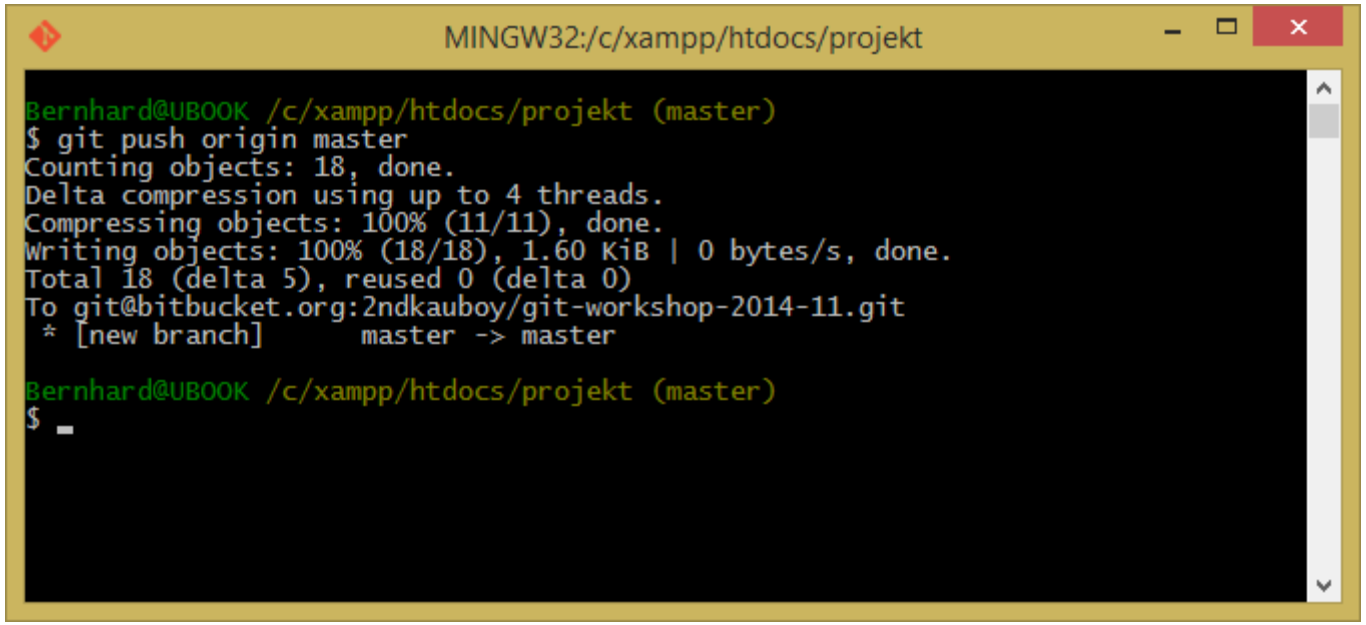
```
git push <remote> <branch>
```



Projekt mit anderen teilen

- Commits an das Remote-Repository schicken

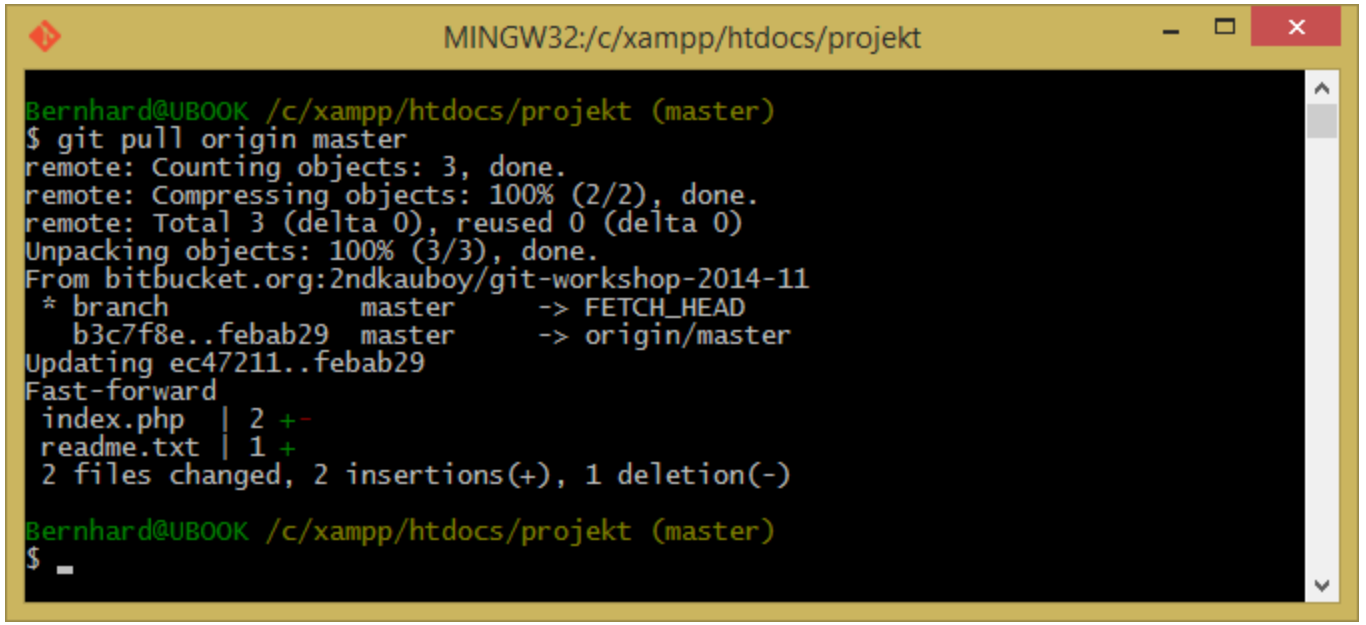
```
git push <remote> <branch>
```



Projekt mit anderen teilen

- Commits aus dem Remote-Repository holen

```
git pull <remote>
```



Das war die Theorie

Let‘s get our handy dirty! ;)

Git einrichten

- Git installieren: http://git-scm.com
- Nutzerdaten ein Konfiguration schreiben:

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

- Texteditor für Commit-Nachrichten einrichten:

```
git config --global "'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst -nosession –noPlugin"
git config --global core.editor 'open -t -n -W'
```

Fragen?

Quellen und gute Ressourcen:

- http://git-scm.com/
- http://rogerdudler.github.io/git-guide/
- https://www.atlassian.com/git/
- https://guides.github.com/
- https://try.github.io/
- http://www.sourcetreeapp.com/