

# git – Eine Einführung

treff.DARC 12.10.2021

Ein Vortrag von DF1HPK  
om@df1hpk.de

# DF1HPK

- Jahrgang 1987
- QRL: Softwareentwickler (Schwerpunkt Java)
- Funkamateurl  
  - 2009 (Klasse E)
  - 2011 (Klasse A)
  - DOK C28
- <https://df1hpk.de>



# Inhalt

- Was ist git
- Die wichtigsten git Befehle
- git Workflows
- Quellen / Weiterführende Informationen

# Was ist git

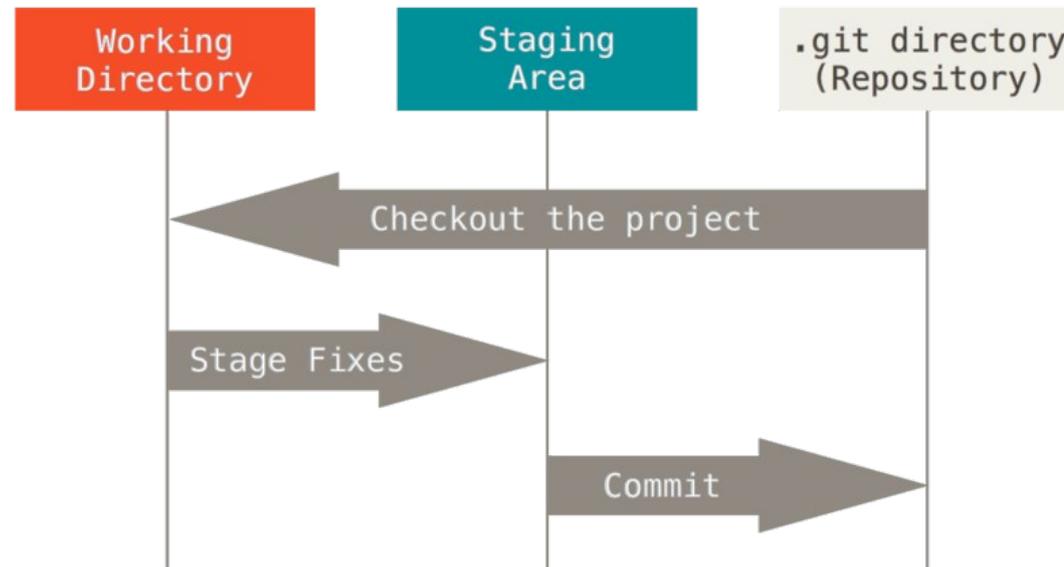
- git ist ein Versionierungstool
  - Es können alle Arten von Dateien versioniert werden
- Initiiert von Linus Torvalds (Erfinder von Linux)
- Frei und Open Source
- Lokal und Server basiert
  - Eigener Server (RaspberryPi / Hosted Server)
  - GitHub / GitLab

# Was ist git

- Kommandozeilen basiert
- Grafische Oberflächen verfügbar
  - TortoiseGit (Windows)
  - gitk (Windows / Linux / MacOS)
  - IDE Erweiterung (z.B. IntelliJ)

# Was ist git

- Drei Hauptzustände einer Datei
  - Modified: Geändert, aber noch nicht versioniert
  - Staged: Für den nächsten Commit (Versionierung) vorgemerkt
  - Comitted: Versionierung gespeichert



Grafik Quelle: <https://git-scm.com/book/en/v2/images/areas.png>

# Die wichtigsten git Befehle

- `git init` – Erstellt neues Repository im aktuellen Ordner
- `git clone` – Clont ein entferntes Repository via HTTP oder SSH
- `git add` – Fügt geänderte Dateien / Ordner zur „Stage“ für den nächsten Commit hinzu
- `git commit` – Erzeugt einen Commit (eine „Version“) der gestageten Dateien
- `git status` – Zeigt an, welche Dateien geändert / gestaged / unbeobachtet sind
- `git diff` – Zeigt Änderungen
- `git log` – Zeigt die gesamte Commit Historie an

# Die wichtigsten git Befehle

- `git restore <datei>` – Verwirft die Änderungen an der Datei
- `git restore --staged <datei>` – Entfernt die Datei aus der Staging-Area, behält aber Änderungen bei.
- `git reset --hard` – Setzt Staging-Area und **alle** Änderungen zurück

# Die wichtigsten git Befehle

- `git branch` – Zeigt alle Branches an
- `git branch <branch>` – Erstellt Branch mit dem angegebenen Namen
- `git checkout <branch>` – Wechselt in den angegebenen Branch
- `git checkout -b <branch>` – Erstellt den angegebenen Branch und wechselt anschließend dorthin
- `git merge <branch>` – Integriert den angegebenen Branch in den aktuellen Branch

# Die wichtigsten git Befehle

- `git remote add` – Erstellt eine neue Verbindung zu einem entfernten Repository
- `git fetch` – Läd alle Referenzen (Branches, Commits etc.) vom entfernten Repository, macht aber lokal **keine** Änderungen am Working Directory
- `git pull` – Läd den aktuellen Branch vom entfernten Repository und **integriert** alle Änderungen in die lokale Kopie
- `git push` – Überträgt den aktuellen Branch (inkl. Commits etc.) ins entfernte Repository

# git Workflows – Neues Repository

## Lokal

```
henrik@miamaria:~$ mkdir vortrag
henrik@miamaria:~$ cd vortrag/
henrik@miamaria:~/vortrag$ git init
Leeres Git-Repository in /home/henrik/vortrag/.git/ initialisiert
henrik@miamaria:~/vortrag$
```

## Server + Clone

```
henrik@miamaria:~$ ssh pi@server
pi@server's password:
Linux server 5.4.83-v7+ #1379 SMP Mon Dec 14 13:08:57 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 12 01:39:06 2021 from 2001:16b8:2d4f:f00:7fd1:c444:d4f0:c329
pi@server:~ $ mkdir repos/vortrag.git
pi@server:~ $ cd repos/vortrag.git/
pi@server:~/repos/vortrag.git $ git init --bare
Leeres Git-Repository in /home/pi/repos/vortrag.git/ initialisiert
pi@server:~/repos/vortrag.git $ exit
Abgemeldet
Connection to server closed.
henrik@miamaria:~$ git clone pi@server:repos/vortrag.git
Klone nach 'vortrag' ...
pi@server's password:
warning: Sie scheinen ein leeres Repository geklont zu haben.
henrik@miamaria:~$
```

# git Workflows – Datei ändern und versionieren

```
henrik@miamaria:~$ cd vortrag/
henrik@miamaria:~/vortrag$ git status
Auf Branch master

Noch keine Commits

nichts zu committen (erstellen/kopieren Sie Dateien und benutzen
Sie "git add" zum Versionieren)
henrik@miamaria:~/vortrag$ nano datei1.txt
henrik@miamaria:~/vortrag$ git status
Auf Branch master

Noch keine Commits

Unversionierte Dateien:
 (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)
    datei1.txt

nichts zum Commit vorgemerkt, aber es gibt unversionierte Dateien
(benutzen Sie "git add" zum Versionieren)
henrik@miamaria:~/vortrag$ git add datei1.txt
henrik@miamaria:~/vortrag$ git status
Auf Branch master

Noch keine Commits

Zum Commit vorgemerkte Änderungen:
 (benutzen Sie "git rm --cached <Datei>..." zum Entfernen aus der Staging-Area)
    neue Datei:    datei1.txt

henrik@miamaria:~/vortrag$ git commit -m "datei1 erstellt"
[master (Root-Commit) 4db2e3a] datei1 erstellt
 1 file changed, 1 insertion(+)
 create mode 100644 datei1.txt
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch basiert auf 'origin/master', aber der Upstream-Branch wurde entfernt.
 (benutzen Sie "git branch --unset-upstream" zum Beheben)

nichts zu committen, Arbeitsverzeichnis unverändert
henrik@miamaria:~/vortrag$ git push -u origin master
pi@server's password:
Objekte aufzählen: 3, fertig.
Zähle Objekte: 100% (3/3), fertig.
Schreibe Objekte: 100% (3/3), 227 Bytes | 227.00 KiB/s, fertig.
Gesamt 3 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
 * [new branch]      master -> master
Branch 'master' folgt nun Remote-Branch 'master' von 'origin'.
henrik@miamaria:~/vortrag$
```

```
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist auf demselben Stand wie 'origin/master'.

nichts zu committen, Arbeitsverzeichnis unverändert
henrik@miamaria:~/vortrag$ nano datei1.txt
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist auf demselben Stand wie 'origin/master'.

Änderungen, die nicht zum Commit vorgemerkt sind:
 (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)
 (benutzen Sie "git restore <Datei>...", um die Änderungen im Arbeitsverzeichnis zu verwerfen)
    geändert:    datei1.txt

keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
henrik@miamaria:~/vortrag$ git add datei1.txt
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist auf demselben Stand wie 'origin/master'.

Zum Commit vorgemerkte Änderungen:
 (benutzen Sie "git restore --staged <Datei>..." zum Entfernen aus der Staging-Area)
    geändert:    datei1.txt

henrik@miamaria:~/vortrag$ git commit -m "Version geändert"
[master b15f457] Version geändert
 1 file changed, 1 insertion(+), 1 deletion(-)
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist 1 Commit vor 'origin/master'.
 (benutzen Sie "git push", um lokale Commits zu publizieren)

nichts zu committen, Arbeitsverzeichnis unverändert
henrik@miamaria:~/vortrag$ git push
pi@server's password:
Objekte aufzählen: 5, fertig.
Zähle Objekte: 100% (5/5), fertig.
Schreibe Objekte: 100% (3/3), 261 Bytes | 261.00 KiB/s, fertig.
Gesamt 3 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
 4db2e3a..b15f457  master -> master
henrik@miamaria:~/vortrag$
```

# git Workflows – fetch / pull

```
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist auf demselben Stand wie 'origin/master'.

nichts zu committen, Arbeitsverzeichnis unverändert
henrik@miamaria:~/vortrag$ git log
commit b15f45724fdde3a9e09457ee6d29b4de6078fb54 (HEAD -> master, origin/master)
Author: Henrik Pfeifer <privat@hpfeifer.de>
Date: Tue Oct 12 01:51:16 2021 +0200

    Version geändert

commit 4db2e3a2f27f83b538beac87977c821aa6ee19f2
Author: Henrik Pfeifer <privat@hpfeifer.de>
Date: Tue Oct 12 01:47:20 2021 +0200

    datei1 erstellt
henrik@miamaria:~/vortrag$ git fetch
pi@server's password:
remote: Objekte aufzählen: 5, Fertig.
remote: Zähle Objekte: 100% (5/5), Fertig.
remote: Gesamt 3 (Delta 0), Wiederverwendet 0 (Delta 0)
Entpacke Objekte: 100% (3/3), 243 Bytes | 60.00 KiB/s, fertig.
Von server:repos/vortrag
 b15f457..79b42b1 master -> origin/master
henrik@miamaria:~/vortrag$ git log
commit b15f45724fdde3a9e09457ee6d29b4de6078fb54 (HEAD -> master)
Author: Henrik Pfeifer <privat@hpfeifer.de>
Date: Tue Oct 12 01:51:16 2021 +0200

    Version geändert

commit 4db2e3a2f27f83b538beac87977c821aa6ee19f2
Author: Henrik Pfeifer <privat@hpfeifer.de>
Date: Tue Oct 12 01:47:20 2021 +0200

    datei1 erstellt
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist 1 Commit hinter 'origin/master', und kann vorgespult werden.
(benutzen Sie "git pull", um Ihren lokalen Branch zu aktualisieren)

nichts zu committen, Arbeitsverzeichnis unverändert
henrik@miamaria:~/vortrag$
```

```
henrik@miamaria:~/vortrag$ git pull
pi@server's password:
Aktualisiere b15f457..79b42b1
Fast-forward
 datei1.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
henrik@miamaria:~/vortrag$ git log
commit 79b42b1439f5475da2df3a8a924d2c8e12be9617 (HEAD -> master, origin/master)
Author: Hans Wurst <hans@wurst.de>
Date: Tue Oct 12 01:56:38 2021 +0200

    changed version (from elsewhere)

commit b15f45724fdde3a9e09457ee6d29b4de6078fb54
Author: Henrik Pfeifer <privat@hpfeifer.de>
Date: Tue Oct 12 01:51:16 2021 +0200

    Version geändert

commit 4db2e3a2f27f83b538beac87977c821aa6ee19f2
Author: Henrik Pfeifer <privat@hpfeifer.de>
Date: Tue Oct 12 01:47:20 2021 +0200

    datei1 erstellt
henrik@miamaria:~/vortrag$
```

# git Workflows - Branches

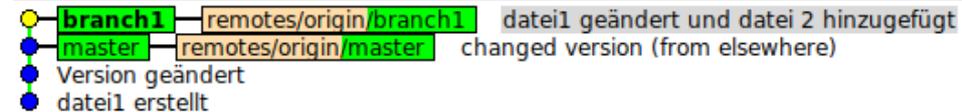
```
henrik@miamaria:~/vortrag$ git branch
* master
henrik@miamaria:~/vortrag$ git checkout -b branch1
Zu neuem Branch 'branch1' gewechselt
henrik@miamaria:~/vortrag$ git branch
* branch1
  master
henrik@miamaria:~/vortrag$ nano datei1.txt
henrik@miamaria:~/vortrag$ nano datei2.txt
henrik@miamaria:~/vortrag$ git status
Auf Branch branch1
Änderungen, die nicht zum Commit vorgemerkt sind:
  (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)
  (benutzen Sie "git restore <Datei>...", um die Änderungen im Arbeitsverzeichnis zu verwerfen)
   geändert:   datei1.txt

Unversionierte Dateien:
  (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)
   datei2.txt

keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
henrik@miamaria:~/vortrag$ git add .
henrik@miamaria:~/vortrag$ git status
Auf Branch branch1
Zum Commit vorgemerkte Änderungen:
  (benutzen Sie "git restore --staged <Datei>..." zum Entfernen aus der Staging-Area)
   geändert:   datei1.txt
  neue Datei:  datei2.txt

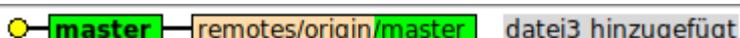
henrik@miamaria:~/vortrag$ git commit -m "datei1 geändert und datei 2 hinzugefügt"
[branch1 4885d44] datei1 geändert und datei 2 hinzugefügt
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 datei2.txt
henrik@miamaria:~/vortrag$ git push -u origin branch1
pi@server's password:
Objekte aufzählen: 6, fertig.
Zähle Objekte: 100% (6/6), fertig.
Delta-Kompression verwendet bis zu 4 Threads.
Komprimiere Objekte: 100% (2/2), fertig.
Schreibe Objekte: 100% (4/4), 333 Bytes | 166.00 KiB/s, fertig.
Gesamt 4 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
 * [new branch]   branch1 -> branch1
Branch 'branch1' folgt nun Remote-Branch 'branch1' von 'origin'.
henrik@miamaria:~/vortrag$
```

```
henrik@miamaria:~/vortrag$ ls -l
insgesamt 8
-rw-rw-r-- 1 henrik henrik 17 Okt 12 18:02 datei1.txt
-rw-rw-r-- 1 henrik henrik 10 Okt 12 18:02 datei2.txt
henrik@miamaria:~/vortrag$ cat datei1.txt
Version 4 branch
henrik@miamaria:~/vortrag$ git checkout master
Zu Zweig »master« gewechselt
Ihr Branch ist auf demselben Stand wie 'origin/master'.
henrik@miamaria:~/vortrag$ ls -l
insgesamt 4
-rw-rw-r-- 1 henrik henrik 10 Okt 12 18:07 datei1.txt
henrik@miamaria:~/vortrag$ cat datei1.txt
Version 3
henrik@miamaria:~/vortrag$ git checkout branch1
Zu Zweig »branch1« gewechselt
Ihr Branch ist auf demselben Stand wie 'origin/branch1'.
henrik@miamaria:~/vortrag$ ls -l
insgesamt 8
-rw-rw-r-- 1 henrik henrik 17 Okt 12 18:07 datei1.txt
-rw-rw-r-- 1 henrik henrik 10 Okt 12 18:07 datei2.txt
henrik@miamaria:~/vortrag$ cat datei1.txt
Version 4 branch
henrik@miamaria:~/vortrag$
```



# git Workflows - Branches

```
henrik@miamaria:~/vortrag$ git branch
branch1
* master
henrik@miamaria:~/vortrag$ nano datei3.txt
henrik@miamaria:~/vortrag$ git add datei3.txt
henrik@miamaria:~/vortrag$ git commit -m "datei3 hinzugefügt"
[master 6ea1272] datei3 hinzugefügt
1 file changed, 1 insertion(+)
create mode 100644 datei3.txt
henrik@miamaria:~/vortrag$ git push
pi@server's password:
Objekte aufzählen: 4, fertig.
Zähle Objekte: 100% (4/4), fertig.
Delta-Kompression verwendet bis zu 4 Threads.
Komprimiere Objekte: 100% (2/2), fertig.
Schreibe Objekte: 100% (3/3), 300 Bytes | 300.00 KiB/s, fertig.
Gesamt 3 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
79b42b1..6ea1272 master -> master
henrik@miamaria:~/vortrag$
```

 **master** — **remotes/origin/master** datei3 hinzugefügt

-  changed version (from elsewhere)
-  Version geändert
-  datei1 erstellt

```
henrik@miamaria:~/vortrag$ ls -l
insgesamt 8
-rw-rw-r-- 1 henrik henrik 10 Okt 12 18:22 datei1.txt
-rw-rw-r-- 1 henrik henrik 17 Okt 12 18:22 datei3.txt
henrik@miamaria:~/vortrag$ git branch
branch1
* master
henrik@miamaria:~/vortrag$ git merge branch1
Merge made by the 'recursive' strategy.
datei1.txt | 2 +-
datei2.txt | 1 +
2 files changed, 2 insertions(+), 1 deletion(-)
create mode 100644 datei2.txt
henrik@miamaria:~/vortrag$ ls -l
insgesamt 12
-rw-rw-r-- 1 henrik henrik 17 Okt 12 18:24 datei1.txt
-rw-rw-r-- 1 henrik henrik 10 Okt 12 18:24 datei2.txt
-rw-rw-r-- 1 henrik henrik 17 Okt 12 18:22 datei3.txt
henrik@miamaria:~/vortrag$ git branch
branch1
* master
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist 2 Commits vor 'origin/master'.
(benutzen Sie "git push", um lokale Commits zu publizieren)

nichts zu committen, Arbeitsverzeichnis unverändert
henrik@miamaria:~/vortrag$ git push
pi@server's password:
Objekte aufzählen: 4, fertig.
Zähle Objekte: 100% (4/4), fertig.
Delta-Kompression verwendet bis zu 4 Threads.
Komprimiere Objekte: 100% (2/2), fertig.
Schreibe Objekte: 100% (2/2), 324 Bytes | 324.00 KiB/s, fertig.
Gesamt 2 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
6ea1272..b723ce1 master -> master
henrik@miamaria:~/vortrag$
```



# git Workflows - Branches

```
henrik@miamaria:~/vortrag$ git branch
branch1
* master
henrik@miamaria:~/vortrag$ nano datei3.txt
henrik@miamaria:~/vortrag$ git add datei3.txt
henrik@miamaria:~/vortrag$ git commit -m "datei3 auf master geändert"
[master df6dabd] datei3 auf master geändert
1 file changed, 1 insertion(+), 1 deletion(-)
henrik@miamaria:~/vortrag$ git push
pi@server's password:
Objekte aufzählen: 5, fertig.
Zähle Objekte: 100% (5/5), fertig.
Delta-Kompression verwendet bis zu 4 Threads.
Komprimiere Objekte: 100% (2/2), fertig.
Schreibe Objekte: 100% (3/3), 285 Bytes | 285.00 KiB/s, fertig.
Gesamt 3 (Delta 1), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
b723ce1..df6dabd master -> master
henrik@miamaria:~/vortrag$ git checkout branch1
Zu Zweig »branch1« gewechselt
Ihr Branch ist auf demselben Stand wie 'origin/branch1'.
henrik@miamaria:~/vortrag$ ls -l
insgesamt 8
-rw-rw-r-- 1 henrik henrik 17 Okt 12 18:24 datei1.txt
-rw-rw-r-- 1 henrik henrik 10 Okt 12 18:24 datei2.txt
henrik@miamaria:~/vortrag$ nano datei3.txt
henrik@miamaria:~/vortrag$ cat datei3.txt
Version 1 branch1
henrik@miamaria:~/vortrag$ git add datei3.txt
henrik@miamaria:~/vortrag$ git commit -m "add datei3"
[branch1 c9f5dbb] add datei3
1 file changed, 1 insertion(+)
create mode 100644 datei3.txt
henrik@miamaria:~/vortrag$ git push
pi@server's password:
Objekte aufzählen: 4, fertig.
Zähle Objekte: 100% (4/4), fertig.
Delta-Kompression verwendet bis zu 4 Threads.
Komprimiere Objekte: 100% (2/2), fertig.
Schreibe Objekte: 100% (3/3), 314 Bytes | 314.00 KiB/s, fertig.
Gesamt 3 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
4885d44..c9f5dbb branch1 -> branch1
henrik@miamaria:~/vortrag$
```

```
henrik@miamaria:~/vortrag$ git branch
* branch1
  master
henrik@miamaria:~/vortrag$ cat datei3.txt
Version 1 branch1
henrik@miamaria:~/vortrag$ git checkout master
Zu Zweig »master« gewechselt
Ihr Branch ist auf demselben Stand wie 'origin/master'.
henrik@miamaria:~/vortrag$ cat datei3.txt
Version 2 Master
henrik@miamaria:~/vortrag$
```

```
henrik@miamaria:~/vortrag$ git merge branch1
KONFLIKT (hinzufügen/hinzufügen): Merge-Konflikt in datei3.txt
automatischer Merge von datei3.txt
Automatischer Merge fehlgeschlagen; beheben Sie die Konflikte und committen Sie dann das Ergebnis.
henrik@miamaria:~/vortrag$
```

```
<<<<<< HEAD
Version 2 Master
=====
Version 1 branch1
>>>>>> branch1
```

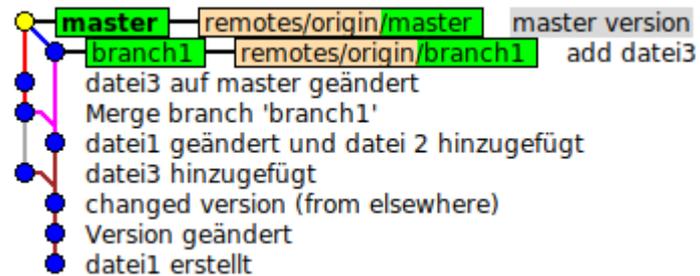
```
henrik@miamaria:~/vortrag$ git status
Auf Branch master
Ihr Branch ist auf demselben Stand wie 'origin/master'.

Sie haben nicht zusammengeführte Pfade.
  (beheben Sie die Konflikte und führen Sie "git commit" aus)
  (benutzen Sie "git merge --abort", um den Merge abzubrechen)

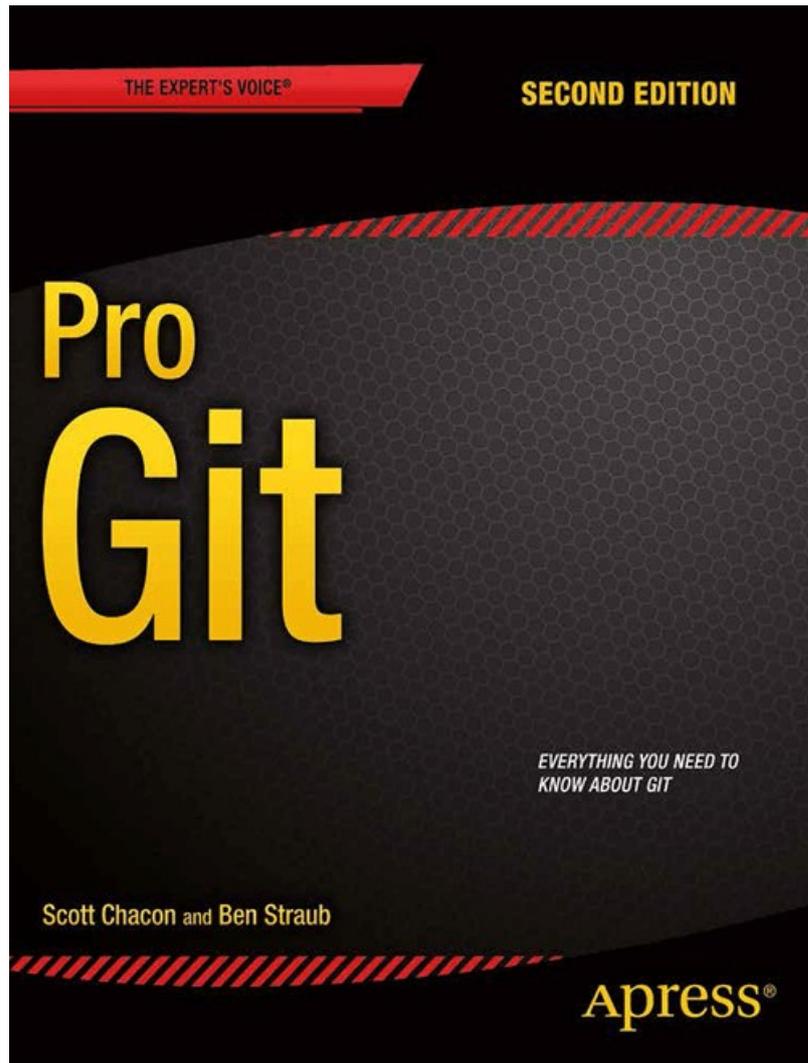
Nicht zusammengeführte Pfade:
  (benutzen Sie "git add/rm <Datei>...", um die Auflösung zu markieren)
  von beiden hinzugefügt: datei3.txt

keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
henrik@miamaria:~/vortrag$ git add datei3.txt
henrik@miamaria:~/vortrag$ git commit -m "master version"
[master b2b878b] master version
henrik@miamaria:~/vortrag$ git push
pi@server's password:
Objekte aufzählen: 1, fertig.
Zähle Objekte: 100% (1/1), fertig.
Schreibe Objekte: 100% (1/1), 215 Bytes | 215.00 KiB/s, fertig.
Gesamt 1 (Delta 0), Wiederverwendet 0 (Delta 0)
To server:repos/vortrag.git
df6dabd..b2b878b master -> master
henrik@miamaria:~/vortrag$ cat datei3.txt
Version 2 Master
henrik@miamaria:~/vortrag$
```

# git Workflows - Branches



# Quellen / Weiterführende Informationen



<https://git-scm.com/book/de/v2>

# Quellen / Weiterführende Informationen

- <https://git-scm.com>
- <https://github.com/>
- <https://about.gitlab.com/>
- [https://de.wikipedia.org/wiki/Liste\\_von\\_Git-GUIs](https://de.wikipedia.org/wiki/Liste_von_Git-GUIs)
- [https://www.atlassian.com/de/dam/jcr:e7e22f25-bba2-4ef1-a197-53f46b6df4a5/SWTM-2088\\_Atlassian-Git-Cheatsheet.pdf](https://www.atlassian.com/de/dam/jcr:e7e22f25-bba2-4ef1-a197-53f46b6df4a5/SWTM-2088_Atlassian-Git-Cheatsheet.pdf)
- <https://git-scm.com/book/de/v2>

# Quellen / Weiterführende Informationen

- Grafiken
  - <https://git-scm.com/book/en/v2/images/areas.png>
  - <https://git-scm.com/images/progit2.png>