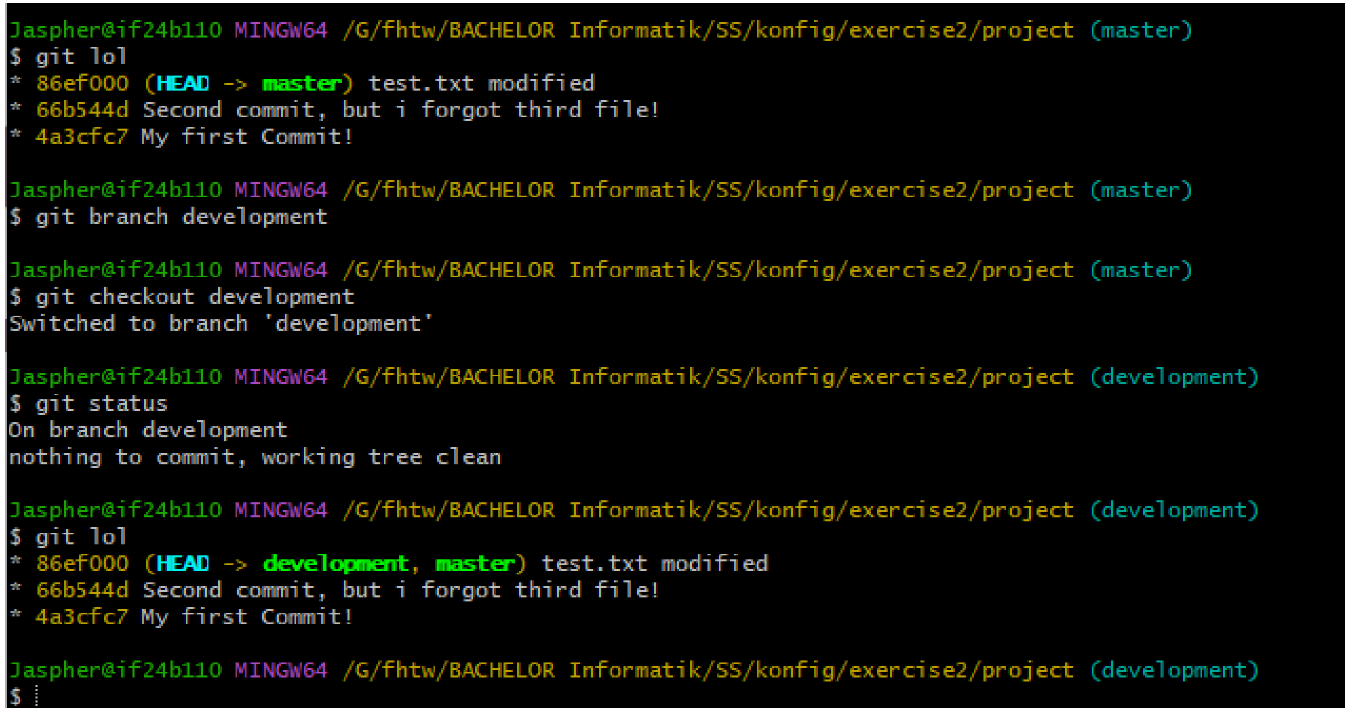
EXERCISE 2

Bif 2

SS 2025

1. BRANCHING / 2. DOCUMENTATION



```
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (master)
$ git lol
* 86ef000 (HEAD -> master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (master)
$ git branch development

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (master)
$ git checkout development
Switched to branch 'development'

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git status
On branch development
nothing to commit, working tree clean

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 86ef000 (HEAD -> development, master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$
```

Variante 1)

- *$git branch <branchname>*
- *$git checkout <branchname>*

Variante 2)

- *$git branch – b <branchname>*

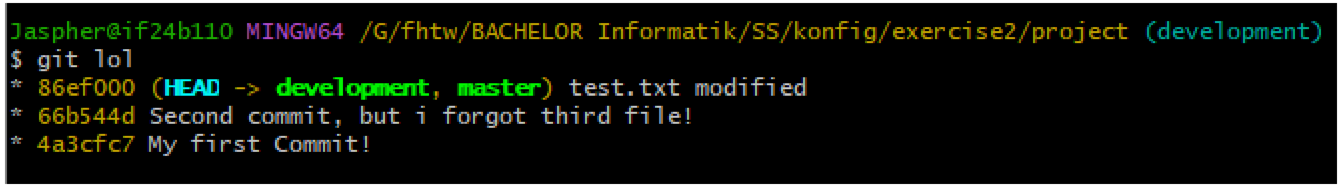
1. BRANCHING / 2. DOCUMENTATION

Variante 1)

- Erstellt ein Branch
- Mittels checkout wechselt HEAD auf genannte Branch
- HEAD zeigt auf <branchname>

Variante 2)

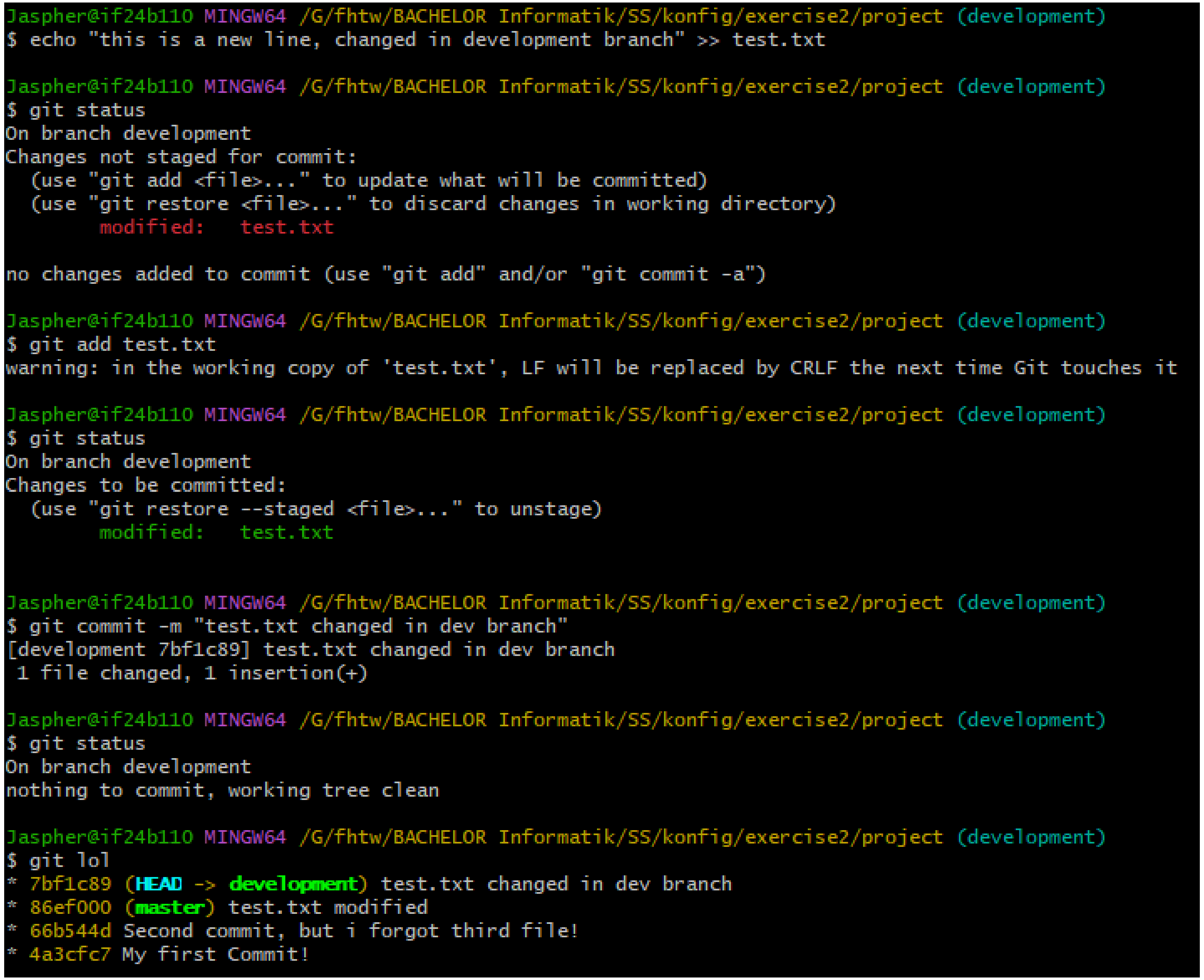
- *$git checkout – b <branchname>*
- -b ist eine Option zum Erstellen eines neuen Branch und zeigt danach auf neu erstellten Branch



```
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 86ef000 (HEAD -> development, master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!
```

3. CHANGES/ 5. COMMIT

- test.txt ueberarbeitet und in die Staging Area gebracht.
- Status zeigt an, dass man sich im Moment auf der development branch befindet.
- Neues commit object erstellt, in diesen Fall auf development
- Commit object auf master liegt darunter als parent



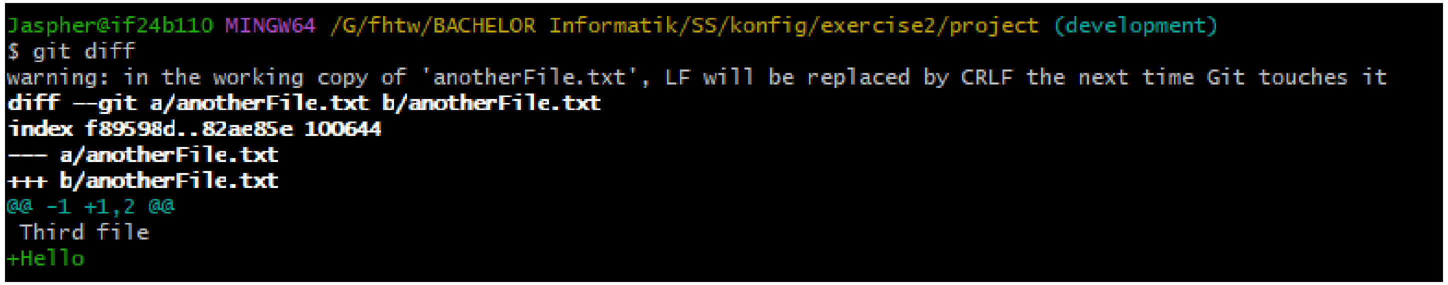
```
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ echo "this is a new line, changed in development branch" >> test.txt

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git status
On branch development
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git add test.txt
warning: in the working copy of 'test.txt', LF will be replaced by CRLF the next time Git touches it

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git status
On branch development
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test.txt


Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git commit -m "test.txt changed in dev branch"
[development 7bf1c89] test.txt changed in dev branch
 1 file changed, 1 insertion(+)

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git status
On branch development
nothing to commit, working tree clean

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 7bf1c89 (HEAD -> development) test.txt changed in dev branch
* 86ef000 (master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!
```

4. DIFF

- *Gibt Auskunft zu Veränderungen der Datei*
- *Name des veraenderten Datei wird angezeigt*
- *Alte Version und neue Version werden angezeigt*
- *Sowie alter/neuer Hashwert*
- *+ … sind Aenderungen*

$git diff

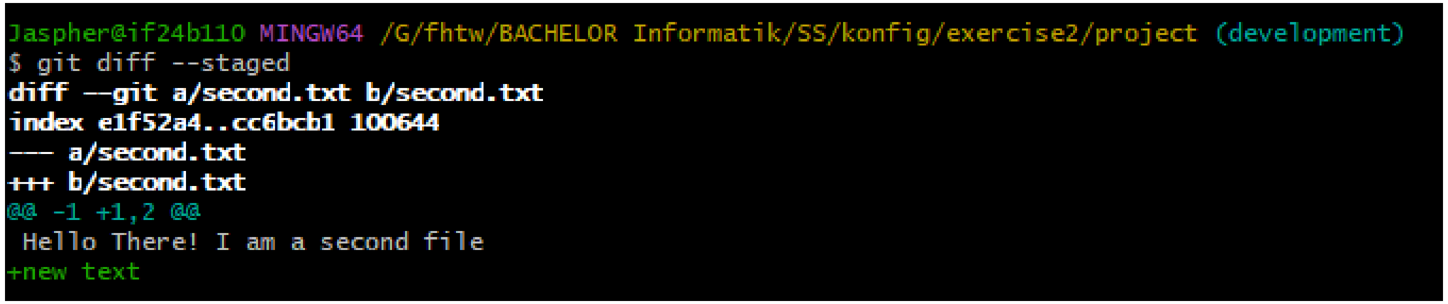
Zeigt Aenderung die noch nicht gestaged sind



```
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git diff
warning: in the working copy of 'anotherFile.txt', LF will be replaced by CRLF the next time Git touches it
diff --git a/anotherFile.txt b/anotherFile.txt
index f89598d..82ae85e 100644
--- a/anotherFile.txt
+++ b/anotherFile.txt
@@ -1 +1,2 @@
 Third file
+Hello
```

$git diff --staged

Aenderung im Staging Area bevor commit



```
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git diff --staged
diff --git a/second.txt b/second.txt
index e1f52a4..cc6bcb1 100644
--- a/second.txt
+++ b/second.txt
@@ -1 +1,2 @@
 Hello There! I am a second file
+new text
```

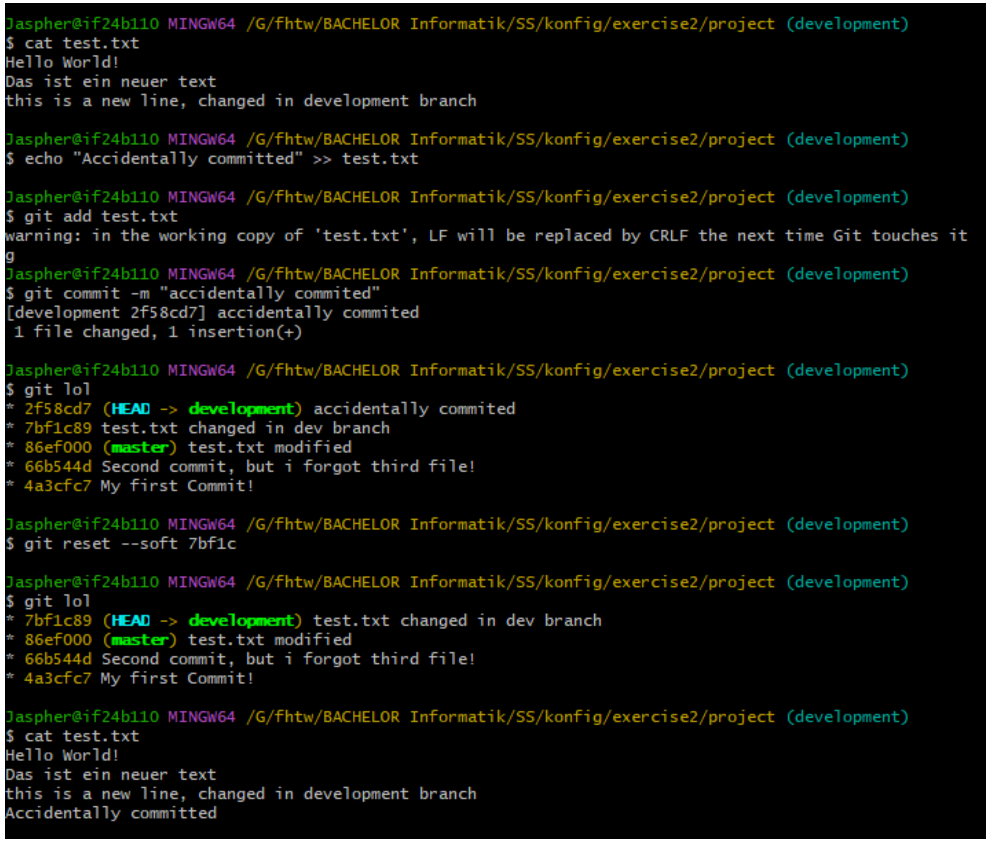
6. ACCIDENTALLY COMMITED

Mittels –soft:

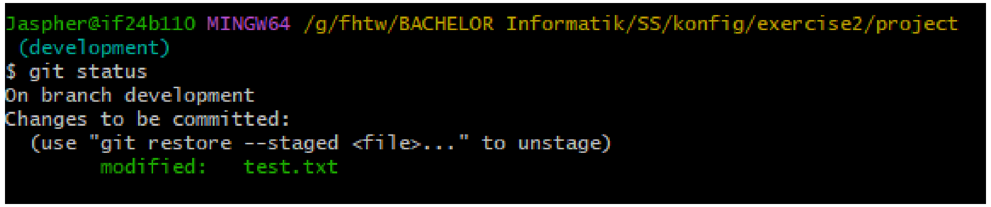
Macht den commit nach [commit object] alles rueckgaengig bis staging area

$git reset [commit object]

Macht alle Commits nach ***[commit object]*** *rueckgaengig*



```
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ cat test.txt
Hello World!
Das ist ein neuer text
this is a new line, changed in development branch

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ echo "Accidentally committed" >> test.txt

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git add test.txt
warning: in the working copy of 'test.txt', LF will be replaced by CRLF the next time Git touches it
g
Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git commit -m "accidentally commited"
[development 2f58cd7] accidentally commited
 1 file changed, 1 insertion(+)

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 2f58cd7 (HEAD -> development) accidentally commited
* 7bf1c89 test.txt changed in dev branch
* 86ef000 (master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git reset --soft 7bf1c

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 7bf1c89 (HEAD -> development) test.txt changed in dev branch
* 86ef000 (master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /G/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ cat test.txt
Hello World!
Das ist ein neuer text
this is a new line, changed in development branch
Accidentally committed
```



```
Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project
 (development)
$ git status
On branch development
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   test.txt
```

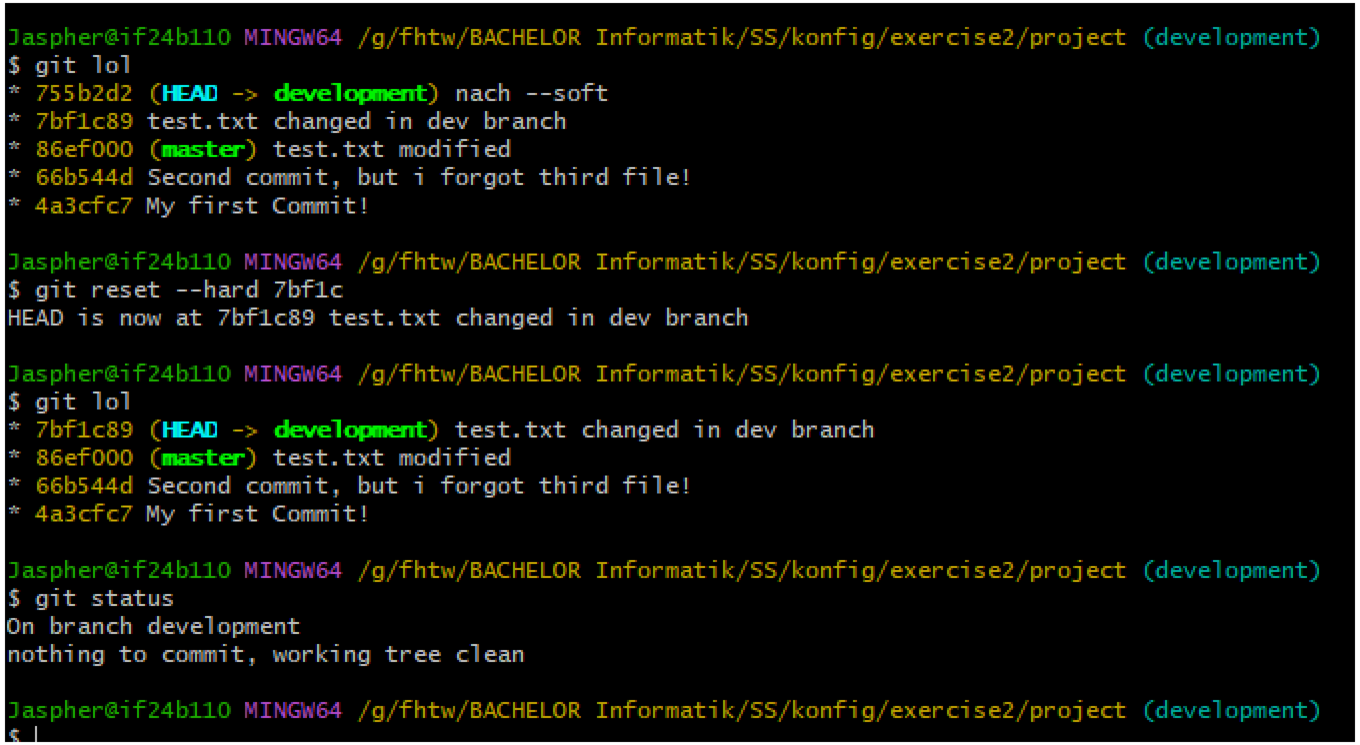
6. ACCIDENTALLY COMMITED

Mittels –hard:

Macht den ganzen commit nach [commit object] rueckgaengig

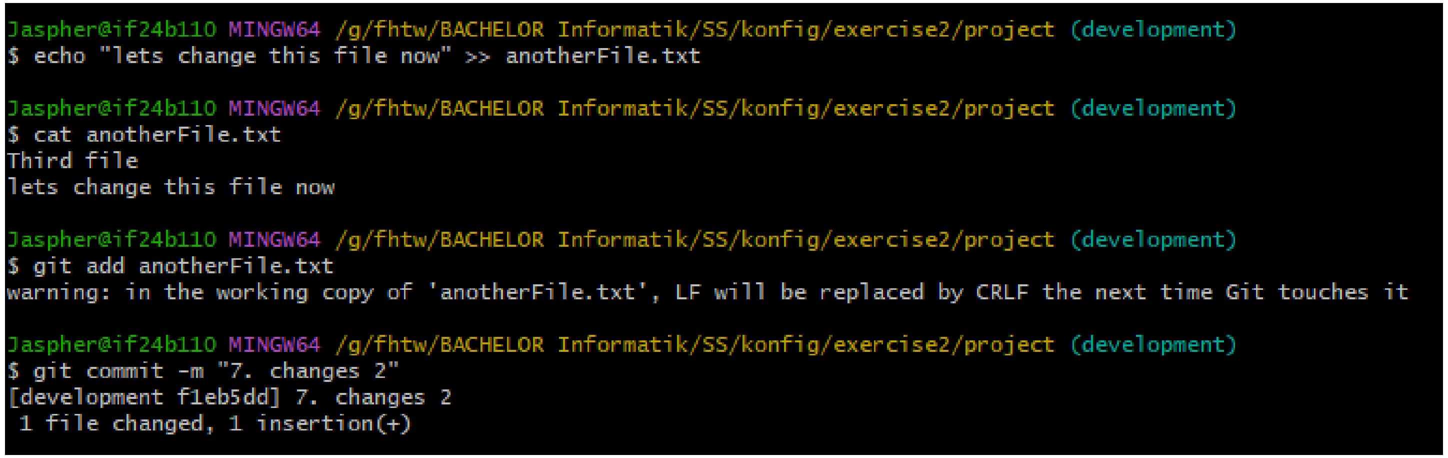
$git reset [commit object]

*Macht alle Commits nach **[commit object]** rueckgaengig*



```
Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 755b2d2 (HEAD -> development) nach --soft
* 7bf1c89 test.txt changed in dev branch
* 86ef000 (master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git reset --hard 7bf1c
HEAD is now at 7bf1c89 test.txt changed in dev branch

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git lol
* 7bf1c89 (HEAD -> development) test.txt changed in dev branch
* 86ef000 (master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git status
On branch development
nothing to commit, working tree clean

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$
```

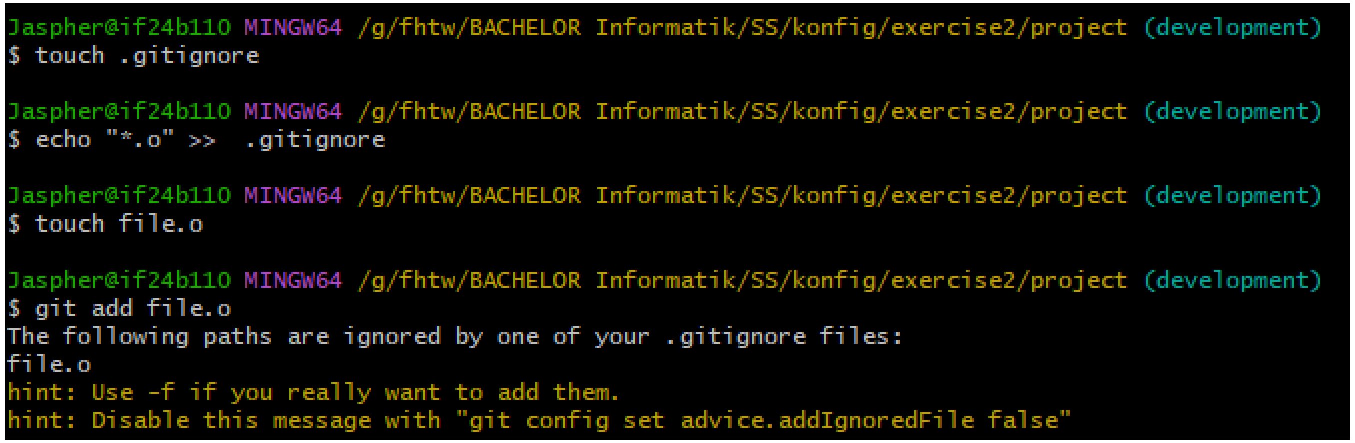
7. CHANGES 2



```
Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ echo "lets change this file now" >> anotherFile.txt

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ cat anotherFile.txt
Third file
lets change this file now

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git add anotherFile.txt
warning: in the working copy of 'anotherFile.txt', LF will be replaced by CRLF the next time Git touches it

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git commit -m "7. changes 2"
[development f1eb5dd] 7. changes 2
 1 file changed, 1 insertion(+)
```

8. .GITIGNORE

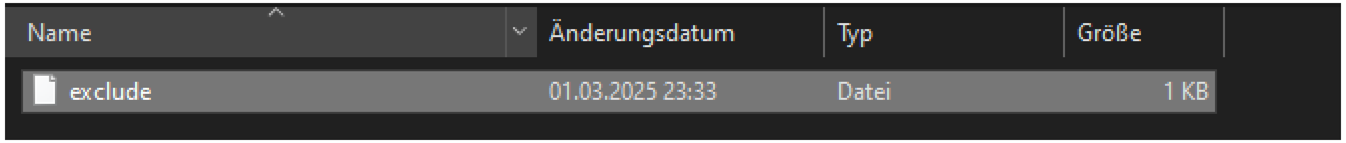
- Eine file .gitignore erstellt
- Mit dem Inhalt: “*.o” befuellt.
- Ein neues File file.o erstellt und versucht in die repository hinzu zufuegen
- Ausgabe sagt: .gitignore ignoriert file.o



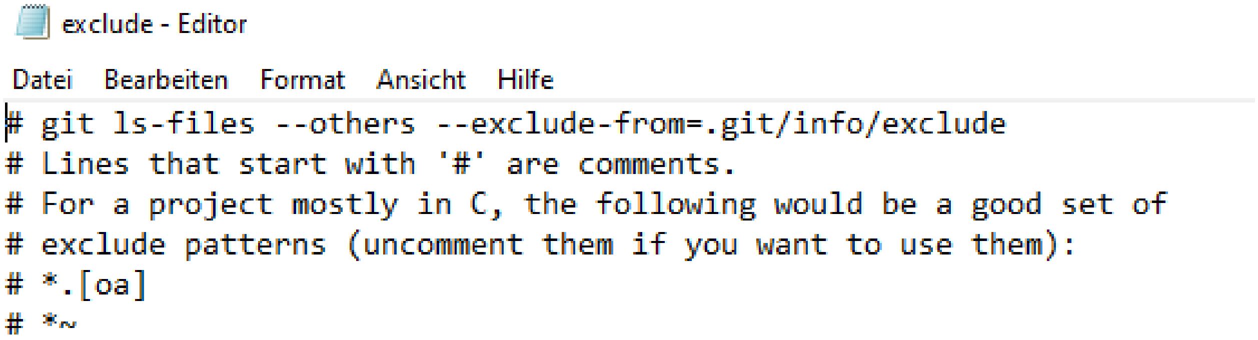
```
Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ touch .gitignore

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ echo "*.o" >>  .gitignore

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ touch file.o

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git add file.o
The following paths are ignored by one of your .gitignore files:
file.o
hint: Use -f if you really want to add them.
hint: Disable this message with "git config set advice.addIgnoredFile false"
```

9. .GIT DIRECTORY

- ../.git/info/exclude



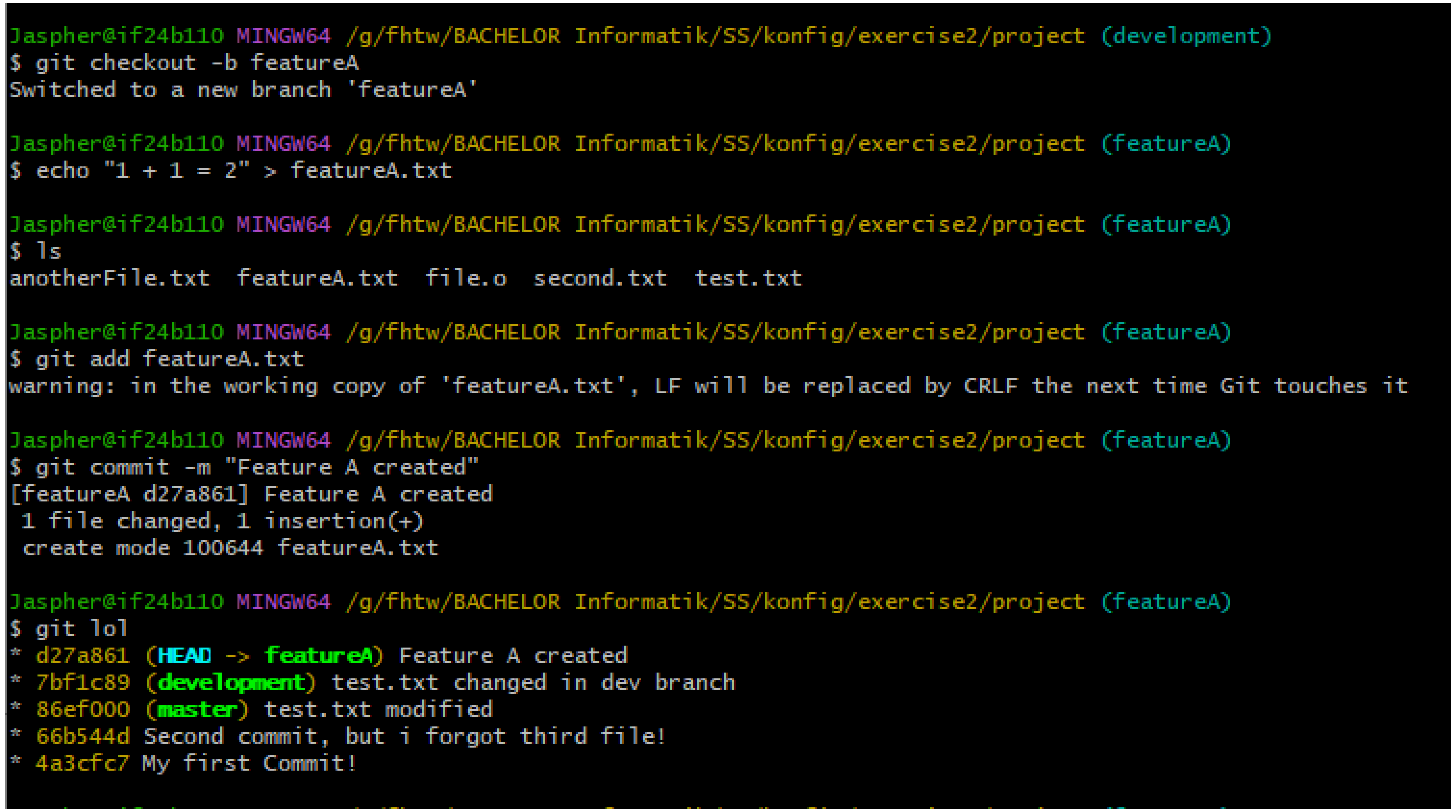
Name	Änderungsdatum	Typ	Größe
exclude	01.03.2025 23:33	Datei	1 KB



```
# git ls-files --others --exclude-from=.git/info/exclude
# Lines that start with '#' are comments.
# For a project mostly in C, the following would be a good set of
# exclude patterns (uncomment them if you want to use them):
# *.[oa]
# *~
```

10. FEATURE BRANCHES

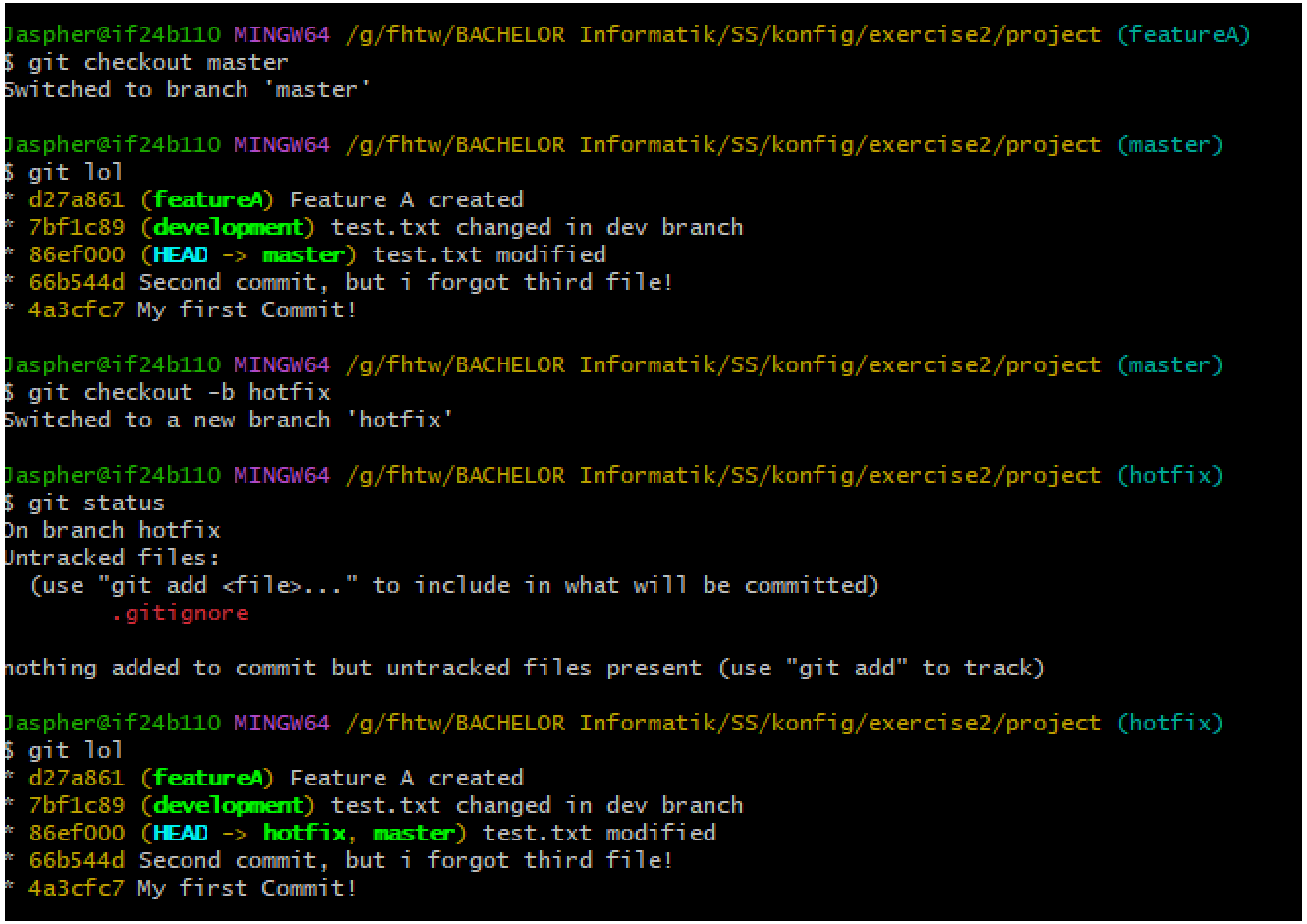
- Ein weiterer branch existiert



```
Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (development)
$ git checkout -b featureA
Switched to a new branch 'featureA'

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (featureA)
$ echo "1 + 1 = 2" > featureA.txt

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (featureA)
$ ls
anotherFile.txt  featureA.txt  file.o  second.txt  test.txt

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (featureA)
$ git add featureA.txt
warning: in the working copy of 'featureA.txt', LF will be replaced by CRLF the next time Git touches it

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (featureA)
$ git commit -m "Feature A created"
[featureA d27a861] Feature A created
 1 file changed, 1 insertion(+)
 create mode 100644 featureA.txt

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (featureA)
$ git lol
* d27a861 (HEAD -> featureA) Feature A created
* 7bf1c89 (development) test.txt changed in dev branch
* 86ef000 (master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!
```

11. HOTFIX

- Zurueck zu master branch gewechselt und ein hotfix branch erstellt.
- HEAD zeigt nun auf hotfix auf das selbe commit, wo master im Moment zeigt



```
Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (featureA)
$ git checkout master
Switched to branch 'master'

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (master)
$ git lol
* d27a861 (featureA) Feature A created
* 7bf1c89 (development) test.txt changed in dev branch
* 86ef000 (HEAD -> master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (master)
$ git checkout -b hotfix
Switched to a new branch 'hotfix'

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (hotfix)
$ git status
On branch hotfix
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)

Jaspher@if24b110 MINGW64 /g/fhtw/BACHELOR Informatik/SS/konfig/exercise2/project (hotfix)
$ git lol
* d27a861 (featureA) Feature A created
* 7bf1c89 (development) test.txt changed in dev branch
* 86ef000 (HEAD -> hotfix, master) test.txt modified
* 66b544d Second commit, but i forgot third file!
* 4a3cfc7 My first Commit!
```