

Homework 3

```
ala@ala-ThinkPad-P50: ~/IdeaProjects/uni/IKT/homework_03/git-homework
File Edit View Search Terminal Help
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit notes.txt
[main (root-commit) b8fe69a] added perplexity overview of git
1 file changed, 18 insertions(+)
create mode 100644 notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano notes.txt

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit notes.txt
[main a83f485] added technical details about git
1 file changed, 10 insertions(+), 3 deletions(-)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git diff
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git log
commit a83f485ac651d317b9a5cd79cf88717e8f13c978 (HEAD -> main)
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 00:10:05 2025 +0200

    added technical details about git

commit b8fe69ad8805a57955fe8cbb5d28c11d5d8c2167
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 00:04:47 2025 +0200

    added perplexity overview of git
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git diff b8fe69ad8805a57955fe8cbb5d28c11d5d8c2167 a83f485
ac651d317b9a5cd79cf88717e8f13c978
diff --git a/notes.txt b/notes.txt
index e1c3d87..859caf6 100644
--- a/notes.txt
+++ b/notes.txt
@@ -1,6 +1,7 @@
-Profil des Open Source-Projekts Git
-Überblick
-Git ist ein freies und quelloffenes, verteiltes Versionskontrollsystem. Es wurde entwickelt, um alles von kleinen bis zu sehr
großen Projekten mit hoher Geschwindigkeit und Effizienz zu verwalten. Git ist heute das am weitesten verbreitete Versionskontrollsystem und bildet das Rückgrat vieler Open Source-Projekte und der modernen Softwareentwicklung.
+Profil des Open Source-Projekts Git
+Überblick Git ist ein freies und quelloffenes, verteiltes Versionskontrollsystem. Es wurde entwickelt, um alles von kleinen bis zu sehr großen
+Projekten mit hoher Geschwindigkeit und Effizienz zu verwalten. Git ist heute das am weitesten verbreitete Versionskontrollsystem und bildet das Rückgrat
+vieler Open Source-Projekte und der modernen Softwareentwicklung.

Hauptmerkmale
Verteiltes System: Jeder Entwickler besitzt eine vollständige Kopie des Repositories inklusive Historie.
@@ -16,3 +17,9 @@ Unterstützung für verschiedene Workflows: Git eignet sich für Einzelentwickler
Geschichte
Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen, nachdem das vorherige System (BitKeeper) nicht mehr genutzt werden konnte.

+Eigenschaft      Beschreibung
+Programmiersprache C, Perl, Shell
+Plattformen      Windows, macOS, Linux, Unix
+Lizenz           GNU General Public License v2 (GPLv2)
+Repository       https://git-scm.com
+Erste Veröffentlichung 2005
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git log
commit a83f485ac651d317b9a5cd79cf88717e8f13c978 (HEAD -> main)

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git log
commit a83f485ac651d317b9a5cd79cf88717e8f13c978 (HEAD -> main)
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 00:10:05 2025 +0200

    added technical details about git

commit b8fe69ad8805a57955fe8cbb5d28c11d5d8c2167
Author: Alexandra Weber <s0574421@htw-berlin.de>
Date: Mon Jul 7 00:04:47 2025 +0200

    added perplexity overview of git
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git status
On branch main
nothing to commit, working tree clean
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$
```

```

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git branch feature
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git switch feature
Switched to branch 'feature'
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano feature.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit feature.txt
error: pathspec 'feature.txt' did not match any file(s) known to git
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git -a commit feature.txt
unknown option: -a
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--config-env=<name>=<envvar>] <command> [<args>]

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add feature.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit feature.txt
[feature 5585f48] added info on git --amend in feature.txt
1 file changed, 40 insertions(+)
create mode 100644 feature.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git switch main
M      notes.txt
Switched to branch 'main'
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git merge feature
Updating a83f485..5585f48
Fast-forward
 feature.txt | 40 +++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 40 insertions(+)
 create mode 100644 feature.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git branch
  feature
* main
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git log --oneline --graph
* 5585f48 (HEAD -> main, feature) added info on git --amend in feature.txt
* a83f485 added technical details about git
* b8fe69a added perplexity overview of git
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$

```

The screenshot displays a code editor window with two panes comparing the contents of a file named 'notes.txt'. The left pane shows the original file content, and the right pane shows the modified content. The changes are highlighted in blue. The editor interface includes a top toolbar with various icons, a search bar, and a status bar at the bottom.

Left Pane (Original Content):

- Profil des Open Source-Projekts Git**
- Überblick**
- Git ist ein freies und quelloffenes, verteiltes Versionskontrollsystem. Es wurde entwickelt, um alles von kleinen Projekten mit hoher Geschwindigkeit und Effizienz zu verwalten. Git ist heute das am weitesten verbreitete Versionenkontrollsystem vieler Open Source-Projekte und der modernen Softwareentwicklung.
- Hauptmerkmale**
- Verteiltes System: Jeder Entwickler besitzt eine vollständige Kopie des Repositories inklusive Historie.
- Effizient und schnell: Git ist für hohe Performance optimiert, sowohl bei kleinen als auch bei großen Projekten.
- Branching und Merging: Lokale Branches sind einfach, schnell und günstig zu erstellen. Das Zusammenführen (Merge) von Branches ist ebenfalls einfach.
- Staging Area: Änderungen können gezielt für den nächsten Commit vorbereitet werden.
- Unterstützung für verschiedene Workflows:** Git eignet sich für Einzelentwickler, kleine Teams und große Open Source-Projekte.
- Geschichte**
- Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen.
- Eigenschaft**
- | Eigenschaft | Beschreibung |
|------------------------|---------------------------------------|
| Programmiersprache | C, Perl, Shell |
| Plattformen | Windows, macOS, Linux, Unix |
| Lizenz | GNU General Public License v2 (GPLv2) |
| Repository | https://git-scm.com |
| Erste Veröffentlichung | 2005 |

Right Pane (Modified Content):

- Profil des Open Source-Projekts Git**
- Überblick**
- Git ist ein freies und quelloffenes, verteiltes Versionskontrollsystem. Es wurde entwickelt, um alles von kleinen Projekten mit hoher Geschwindigkeit und Effizienz zu verwalten. Git ist heute das am weitesten verbreitete Versionenkontrollsystem vieler Open Source-Projekte und der modernen Softwareentwicklung.
- Hauptmerkmale**
- Verteiltes System: Jeder Entwickler besitzt eine vollständige Kopie des Repositories inklusive Historie.
- Effizient und schnell: Git ist für hohe Performance optimiert, sowohl bei kleinen als auch bei großen Projekten.
- Branching und Merging: Lokale Branches sind einfach, schnell und günstig zu erstellen. Das Zusammenführen (Merge) von Branches ist ebenfalls einfach.
- Staging Area: Änderungen können gezielt für den nächsten Commit vorbereitet werden.
- Unterstützung für verschiedene Workflows:** Git eignet sich für Einzelentwickler, kleine Teams und große Open Source-Projekte.
- Geschichte**
- Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen.
- Eigenschaft**
- | Eigenschaft | Beschreibung |
|------------------------|---------------------------------------|
| Programmiersprache | C, Perl, Shell |
| Plattformen | Windows, macOS, Linux, Unix |
| Lizenz | GNU General Public License v2 (GPLv2) |
| Repository | https://git-scm.com |
| Erste Veröffentlichung | 2005 |

```
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git checkout -b experiment
Switched to a new branch 'experiment'
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit notes.txt
[experiment b4da62a] changed layout in notes.txt
 1 file changed, 18 insertions(+), 10 deletions(-)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git switch main
Switched to branch 'main'
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit notes.txt
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    branch_merge.png
    diff_log.png
    log_status.png

nothing added to commit but untracked files present (use "git add" to track)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit notes.txt
[main 5ec2a87] changed some details in notes.txt
 1 file changed, 6 insertions(+), 1 deletion(-)
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git merge experiment
Auto-merging notes.txt
CONFLICT (content): Merge conflict in notes.txt
Automatic merge failed; fix conflicts and then commit the result.
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git meld
git: 'meld' is not a git command. See 'git --help'.

The most similar command is
    help
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ meld notes.txt
^Cala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git merge experiment
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
```

```

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git diff
diff --cc notes.txt
index 62e5428,5947740..0000000
--- a/notes.txt
+++ b/notes.txt
@@@ -12,19 -12,22 +12,23 @@@ Verteiltes System: Jeder Entwickler bes

Effizient und schnell: Git ist für hohe Performance optimiert, sowohl bei kleinen als auch bei großen Projekten.

--Branching und Merging: Lokale Branches sind einfach, schnell und günstig zu erstellen. Das Zusammenführen (Merging) von Branches ist ein zentrales Feature.
++Branching und Merging: Lokale Branches sind einfach, schnell und günstig zu erstellen.
++Das Zusammenführen (Merging) von Branches ist ein zentrales Feature.

Staging Area: Änderungen können gezielt für den nächsten Commit vorbereitet werden.

- Unterstützung für verschiedene Workflows: Git eignet sich für Einzelentwickler, kleine Teams und große Open Source-Projekte gleichermaßen.
-Unterstützung für verschiedene Workflows: Git eignet sich für Einzelentwickler, kleine Teams und große Open Source-Projekte gleichermaßen
++Unterstützung für verschiedene Workflows: Git eignet sich für Einzelentwickler, kleine Teams und große Open
++Source-Projekte gleichermaßen.

Geschichte
~~~~~
+-----
Entstehung: Git wurde 2005 von Linus Torvalds entwickelt, um die Entwicklung des Linux-Kernels zu unterstützen, nachdem das vorherige System (BitKeeper) nicht mehr genutzt werden konnte.

- Eigenschaft          Beschreibung
- Programmiersprache   C, Perl, Shell
- Plattformen          Windows, macOS, Linux, Unix
- Lizenz               GNU General Public License v2 (GPLv2)
- Repository           https://git-scm.com
- Erste Veröffentlichung 2005
+-----
- Eigenschaft          |Beschreibung|
++Eigenschaft          |Beschreibung|
+-----
+ Programmiersprache   |C, Perl, Shell|
+ Plattformen          |Windows, macOS, Linux, Unix|
+ Lizenz               |GNU General Public License v2 (GPLv2)|
+ Repository           |https://git-scm.com|
+ Erste Veröffentlichung |2005|
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ nano notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git switch experiment ; nano notes.txt
fatal: cannot switch branch while merging
Consider "git merge --quit" or "git worktree add".

```

```

ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git rm notes.txt _*
fatal: pathspec '_*' did not match any files
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git rm notes_*
fatal: pathspec 'notes_BACKUP_12932.txt' did not match any files
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ ll
total 564
drwxrwxr-x 3 ala ala 4096 Jul 7 00:55 ./
drwxrwxr-x 3 ala ala 4096 Jul 7 00:01 ../
-rw-rw-r-- 1 ala ala 150471 Jul 7 00:34 branch_merge.png
-rw-rw-r-- 1 ala ala 251418 Jul 7 00:15 diff_log.png
-rw-rw-r-- 1 ala ala 2316 Jul 7 00:32 feature.txt
drwxrwxr-x 8 ala ala 4096 Jul 7 00:58 .git/
-rw-rw-r-- 1 ala ala 57845 Jul 7 00:17 log_status.png
-rw-rw-r-- 1 ala ala 1631 Jul 7 00:53 notes_BACKUP_12932.txt
-rw-rw-r-- 1 ala ala 1445 Jul 7 00:53 notes_BASE_12932.txt
-rw-rw-r-- 1 ala ala 1616 Jul 7 00:55 notes_LOCAL_12932.txt
-rw-rw-r-- 1 ala ala 1616 Jul 7 00:55 notes_REMOTE_12932.txt
-rw-rw-r-- 1 ala ala 1616 Jul 7 00:55 notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git rm notes_BACKUP_12932.txt notes_BASE_12932.txt notes_
LOCAL_12932.txt notes_REMOTE_12932.txt
fatal: pathspec 'notes_BACKUP_12932.txt' did not match any files
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ ll
total 564
drwxrwxr-x 3 ala ala 4096 Jul 7 00:55 ./
drwxrwxr-x 3 ala ala 4096 Jul 7 00:01 ../
-rw-rw-r-- 1 ala ala 150471 Jul 7 00:34 branch_merge.png
-rw-rw-r-- 1 ala ala 251418 Jul 7 00:15 diff_log.png
-rw-rw-r-- 1 ala ala 2316 Jul 7 00:32 feature.txt
drwxrwxr-x 8 ala ala 4096 Jul 7 00:59 .git/
-rw-rw-r-- 1 ala ala 57845 Jul 7 00:17 log_status.png
-rw-rw-r-- 1 ala ala 1631 Jul 7 00:53 notes_BACKUP_12932.txt
-rw-rw-r-- 1 ala ala 1445 Jul 7 00:53 notes_BASE_12932.txt
-rw-rw-r-- 1 ala ala 1616 Jul 7 00:55 notes_LOCAL_12932.txt
-rw-rw-r-- 1 ala ala 1616 Jul 7 00:55 notes_REMOTE_12932.txt
-rw-rw-r-- 1 ala ala 1616 Jul 7 00:55 notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git branch
experiment
feature
* main
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git add notes.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit notes.txt
fatal: cannot do a partial commit during a merge.
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git commit
[main 73d3bba] Merge branch 'experiment' fixed all conflicts with meld
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ meld notes.txt notes_
notes_BACKUP_12932.txt notes_BASE_12932.txt notes_LOCAL_12932.txt notes_REMOTE_12932.txt
ala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ meld notes.txt notes_BASE_12932.txt
^Cala@ala-ThinkPad-P50:~/IdeaProjects/uni/IKT/homework_03/git-homework$ git log --oneline --graph
* 73d3bba (HEAD -> main) Merge branch 'experiment' fixed all conflicts with meld
| \
| * b4da62a (experiment) changed layout in notes.txt
| * | 5ec2a87 changed some details in notes.txt
| /
* 5585f48 (feature) added info on git --amend in feature.txt
* a83f485 added technical details about git
* b8fe69a added perplexity overview of git

```

Probleme:

Ich habe lange nicht verstanden, dass es nur um git commit, ohne Angabe einer Datei geht.

Fragen:

1

einfacher/fast forward merge

- aktueller Branch ist direkter Vorfahre, keine parallele Entwicklung (an denselben Dateien)
- der aktuelle Branch endet mit dem neu hineingemergten commit
- 3-way merge
- Git sucht gemeinsamen Vorfahren (letzter gemeinsamer Stand -> Base)
- Vergleich von
 1. Base
 2. HEAD = aktueller Branch, akt. Stand
 3. Source = Branch, der gemerged werden soll, akt. Stand
- beim merge werden die Entwicklungen zusammengeführt:

```

(main) A---B---C---F
          \

```

- der merge erzeugt einen neuen commit (hier G)
- dieser hat zwei Vorfahren

2

In einem Versionskontrollsystem werden jedewede Änderungen an Dateien erfasst, dokumentiert und verwaltet. So kann jederzeit nachvollzogen werden, wann was von wem geändert wurde (Zeitstempel, Benutzerkennung).

Änderungen können rückgängig bzw. ein früherer Zustand wiederhergestellt werden.

Das ermöglicht die gemeinsame Arbeit an Projekten.

Probleme/Konflikte werden durchs System erkannt.

Jede Version des Projekts wird gespeichert.

3

Getrackte, geänderte Dateien werden beim commit als snapshot gespeichert. Alle unveränderten Dateien werden mittels hashes -> Verweisen auf existierende Objekte repräsentiert.

Jeder commit erhält eine eindeutige SHA1-ID.

SHA-Hashes ergeben sich aus dem Secure Hash Algorithm 1, einer kryptographischen Funktion, die aus Eingaben beliebiger Länge 160 Bit lange Werte berechnet (Darstellung als Hexadezimalzahl, 40 Stellen).

Eine identische Eingabe führt jedes Mal zum selben Hash-Wert.

Da auch die unveränderten Dateien lediglich durch Hash-Verweise auf ihre Vorfahren gespeichert werden, ist Git extrem effizient. diffs können zwar aufgerufen werden, werden aber nicht gespeichert.

4

Branching ermöglicht die Arbeit an unterschiedlichen Funktionen, ggf. durch verschiedene Entwickelnde, zur gleichen Zeit. Indem von einem Zustand aus ein Branch erstellt wird, wird das Gesamtprojekt zunächst nicht verändert.

Beispiel:

Bei der Entwicklung einer Website kümmert sich Alice um die Login-Fkt., Bülent ums CMS und 3 weitere um die Datenbank-API. Bereits bestehende Komponenten bleiben unberührt, bis ein Branch ins Hauptprojekt gemerged wird. Ob dann noch alles funktioniert, kann auf einem Test-Branch ausprobiert werden.

Andere Use Cases:

Es sollen verschiedene Designs getestet werden.

Es gibt Ideen für neue Funktionen und unterschiedliche Ansätze zur Umsetzung.