

A close-up photograph of a highly polished chrome air cleaner mounted on a V8 engine. The air cleaner has a large, ribbed top cover and a prominent Edelbrock badge. The engine block and surrounding components are also made of chrome, reflecting light. The background is dark, making the chrome stand out.

Git verstehen

Unter der Haube der Quellcodeverwaltung...



Michael Kaufmann

Vice President Consulting Service



20 Jahre Softwareentwickler
15 Jahre ALM & DevOps
8 Jahre Git
Microsoft Regional Director & MVP

@mike_kaufmann

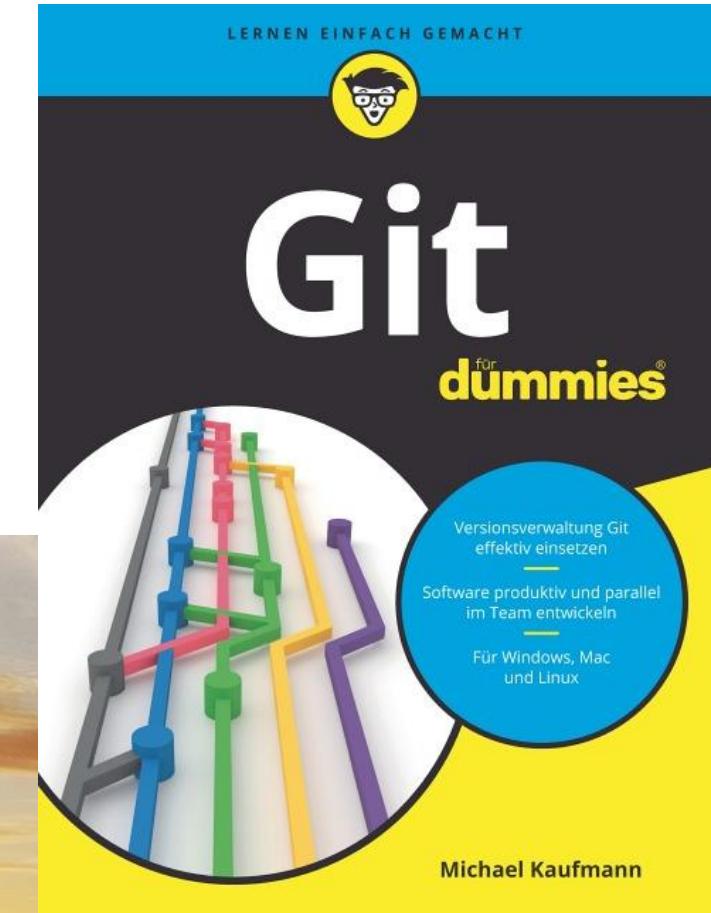
@wulfland

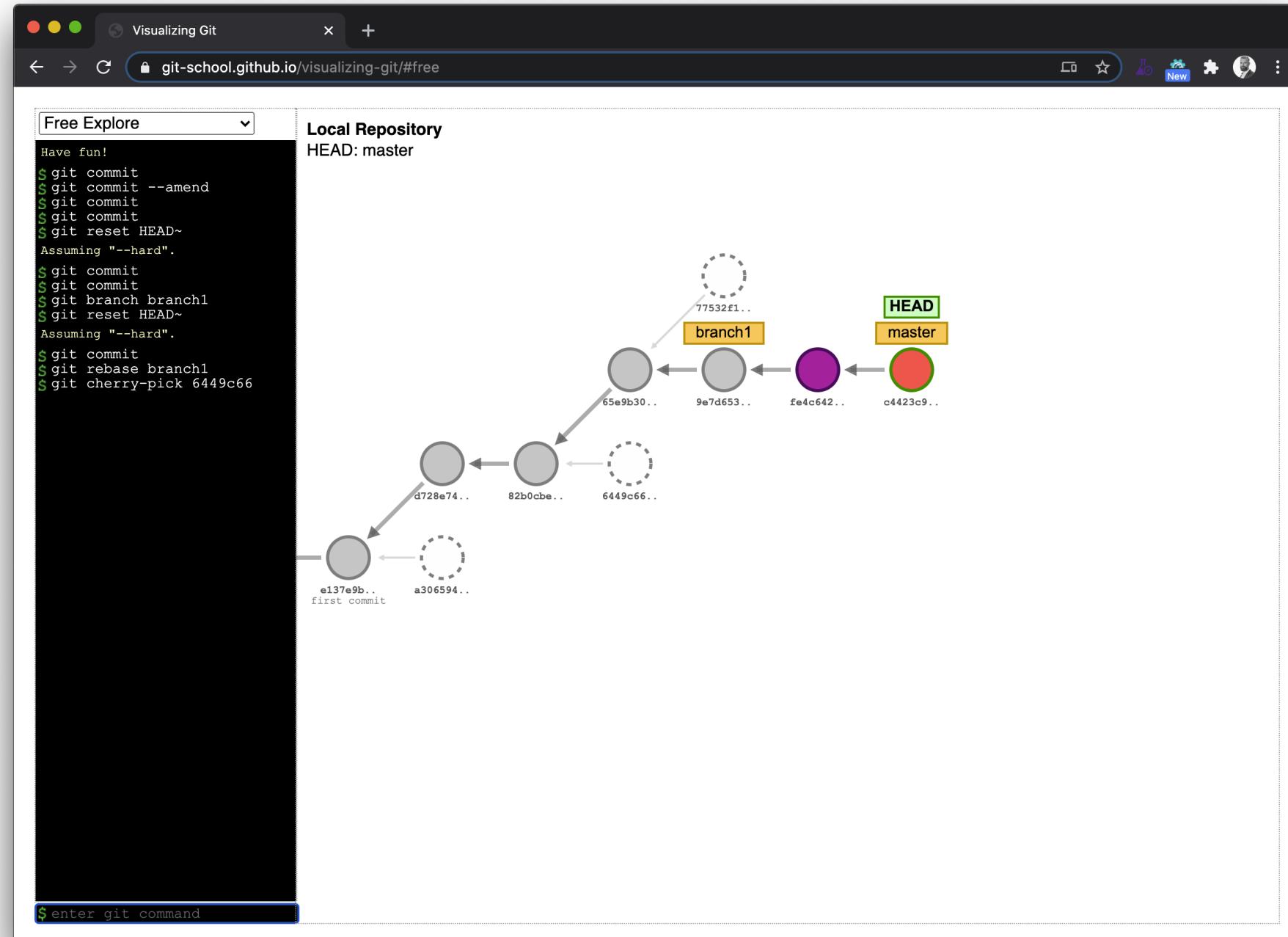
<https://writeabout.net>

Git für Dummies, Michael Kaufmann, Wiley-VCH, 2021



21. Januar 2021





THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



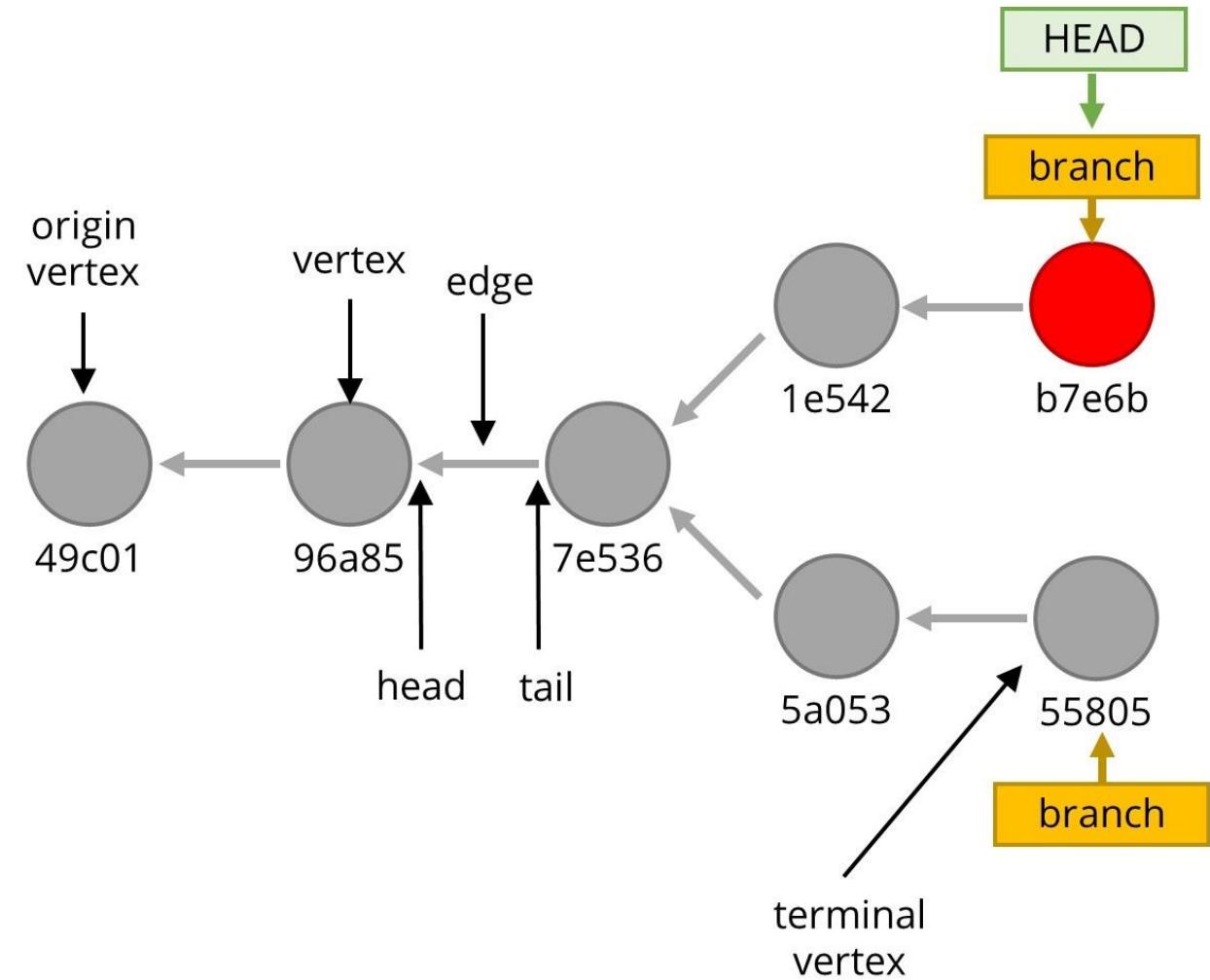
Was ist Git?

»Das ist Git. Es hilft bei der Nachverfolgung der kollaborativen Zusammenarbeit in Projekten durch die Nutzung eines wunderschönen verteilten Baums der Graphentheorie.«

»Cool. Wie verwendet man es?«

»Keine Ahnung. Lerne einfach die folgenden Befehle auswendig und gib sie auf der Kommandozeile ein, um deine Arbeit zu synchronisieren. Wenn du auf Fehler stößt, dann speichere deine Arbeit woanders, lösche das Projekt und lade eine neue Kopie herunter.«

Der gerichtete azyklische Graph (DAG)



Der HASH

Ein *HASH* ist ein kryptographischer Prüfwert zur Integritätsprüfung von unter anderem Dateien. Durch Vergleich des HASH-Wertes kann man feststellen, ob sich eine Datei geändert hat. Eine Datei ergibt immer nur denselben HASH, wenn Sie nicht geändert wurde.

Der SHA-1

Bei *SHA-1* handelt es sich um ***Secure Hash Algorithm*** – eine Gruppe standardisierter kryptologischer Hashfunktionen. Sie dienen zur Berechnung von Prüfwerten für digitale Daten. Der ursprüngliche SHA wurde wegen eines »Konstruktionsfehlers« schon 1995 korrigiert und spielte deswegen keine Rolle mehr. Er ist heute als SHA-0 bekannt, die korrigierte Variante als SHA-1.

Was ist ein Commit?





Warnung

Viele Kommandos hier sind nicht
für den täglichen Gebrauch
gedacht!

Demo: Objects and Trees



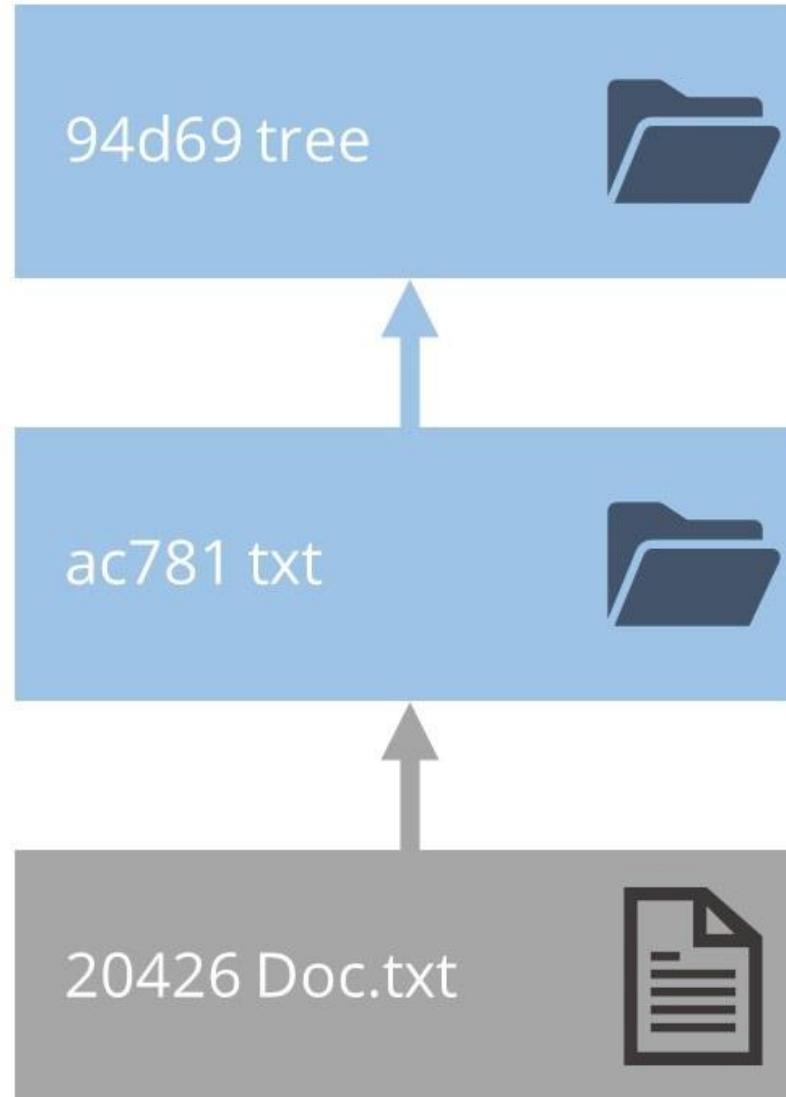
```
..UnterDerHaube (zsh) ~source/repos/basta/UnterDerHaube 12:59:34 ⓘ
> git init
Initialized empty Git repository in /Users/m.kaufmann/source/repos/basta/UnterDerHaube/.git/
Apple ~source/repos/basta/UnterDerHaube git ⌂ main ?1 12:59:37 ⓘ
> git hash-object -w txt/Dox.txt
f20acb769ad757b2058c8c3c96e7eac2e1258503
Apple ~source/repos/basta/UnterDerHaube git ⌂ main ?1 12:59:47 ⓘ
> cat .git/objects/f2/0acb769ad757b2058c8c3c96e7eac2e1258503
xK00R00`0K0M0Rp)0Sp0M-QPr0VRp*0/000/0L00K0Q(0)2000040%
Apple ~source/repos/basta/UnterDerHaube git ⌂ main ?1 12:59:59 ⓘ
> git cat-file -p f20ac
Name: Dr. Emmet "Doc" Brown
Original time: 1985
Apple ~source/repos/basta/UnterDerHaube git ⌂ main ?1 13:00:12 ⓘ
> git update-index --add --cacheinfo 10644 f20acb769ad757b2058c8c3c96e7eac2e1258503 tx
t/Dox.txt
Apple ~source/repos/basta/UnterDerHaube git ⌂ main +1 !1 13:01:06 ⓘ
> git ls-files --stage
100644 f20acb769ad757b2058c8c3c96e7eac2e1258503 0      txt/Dox.txt
Apple ~source/repos/basta/UnterDerHaube git ⌂ main +1 !1 13:01:35 ⓘ
> git write-tree
409d1da9e721d43e6e42c5fee167e8ba36d5133a
Apple ~source/repos/basta/UnterDerHaube git ⌂ main +1 !1 13:02:08 ⓘ
> git ls-tree 409d
040000 tree 4590a439fb3308d87d6d8f8d9b34f55e9fa6a11a      txt
Apple ~source/repos/basta/UnterDerHaube git ⌂ main +1 !1 13:02:43 ⓘ
> git ls-tree 4590
100644 blob f20acb769ad757b2058c8c3c96e7eac2e1258503      Dox.txt
Apple ~source/repos/basta/UnterDerHaube git ⌂ main +1 !1 13:02:54 ⓘ
>
```

Every 2.0s: tree -a

```
Michaels-MacBook-Pro.local: Thu Sep 17 13:03:22 2020
```

```
.git
├── HEAD
├── config
├── description
└── hooks
    ├── applypatch-msg.sample
    ├── commit-msg.sample
    ├── fsmonitor-watchman.sample
    ├── post-update.sample
    ├── pre-applypatch.sample
    ├── pre-commit.sample
    ├── pre-merge-commit.sample
    ├── pre-push.sample
    ├── pre-rebase.sample
    ├── pre-receive.sample
    └── prepare-commit-msg.sample
        └── update.sample
├── index
├── info
│   └── exclude
└── objects
    ├── 40
    │   └── 9d1da9e721d43e6e42c5fee167e8ba36d5133a
    ├── 45
    │   └── 90a439fb3308d87d6d8f8d9b34f55e9fa6a11a
    └── f2
        └── 0acb769ad757b2058c8c3c96e7eac2e1258503
    └── info
    └── pack
└── refs
    ├── heads
    └── tags
└── txt
    └── Dox.txt
```

13 directories, 21 files



Der Aufbau des tree

Demo: Commit und Ref



```

❯ git commit-tree 409d1da9e721d43e6e42c5fee167e8ba36d5133a -m "Hallo Basta"
d7b9f5205e34558ede629149e27772c2101a6f4f
❯ git status
On branch main

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  txt/Dox.txt

❯ git cat-file -p d7b9f
tree 409d1da9e721d43e6e42c5fee167e8ba36d5133a
author Michael Kaufmann <5276337+wulfland@users.noreply.github.com> 1600341694 +0200
committer Michael Kaufmann <5276337+wulfland@users.noreply.github.com> 1600341694 +0200

Hallo Basta
❯ git update-ref refs/heads/main d7b9f5205e34558ede629149e27772c2101a6f4f
❯ git status
On branch main
nothing to commit, working tree clean
❯

```

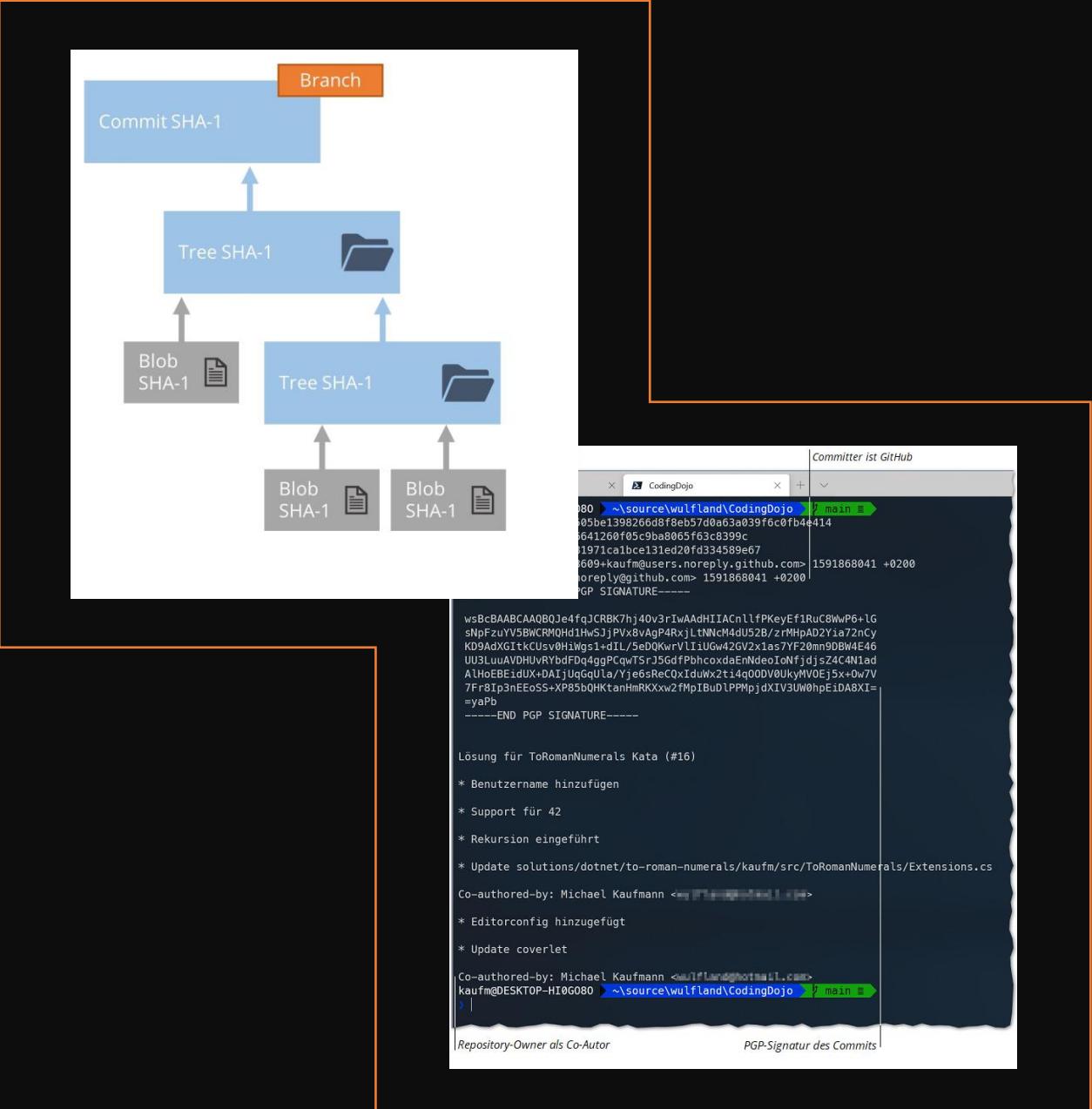
```

Every 2.0s: tree -a
Michaels-MacBook-Pro.local: Thu Sep 17 13:23:20 2020

.
+-- .git
    |-- HEAD
    |-- config
    |-- description
    |-- hooks
        |-- applypatch-msg.sample
        |-- commit-msg.sample
        |-- fsmonitor-watchman.sample
        |-- post-update.sample
        |-- pre-applypatch.sample
        |-- pre-commit.sample
        |-- pre-merge-commit.sample
        |-- pre-push.sample
        |-- pre-rebase.sample
        |-- pre-receive.sample
        |-- prepare-commit-msg.sample
        |-- update.sample
    |-- index
    |-- info
        |-- exclude
    |-- logs
        |-- HEAD
    |-- refs
        |-- heads
            |-- main
    |-- objects
        |-- 40
            |-- 9d1da9e721d43e6e42c5fee167e8ba36d5133a
        |-- 45
            |-- 90a439fb3308d87d6d8f8d9b34f55e9fa6a11a
        |-- 7d
            |-- dc5a21ba0210931c432b5589102a04c2fd17cf
        |-- d7
            |-- b9f5205e34558ede629149e27772c2101a6f4f
        |-- f2
            |-- 0acb769ad757b2058c8c3c96e7eac2e1258503
        |-- info
        |-- pack
    |-- refs
        |-- heads
            |-- main
        |-- tags

```

Aufbau des Commits



```

❯ ~source/repos/basta/UnterDerHaube git ⌂ main ..... 13:31:31 ⓘ
❯ echo "Change" >> txt/Dox.txt
❯ ~source/repos/basta/UnterDerHaube git ⌂ main !1 ..... 13:31:45 ⓘ
❯ git add .
❯ ~source/repos/basta/UnterDerHaube git ⌂ main +1 ..... 13:31:51 ⓘ
❯ git commit -m "Normal change"
[main 124b36b] Normal change
 1 file changed, 1 insertion(+)
❯ ~source/repos/basta/UnterDerHaube git ⌂ main ..... 13:32:05 ⓘ
❯ git log -1
commit 124b36b0d18dbe356e30442105bf7529ebc3e971 (HEAD -> main)
Author: Michael Kaufmann <5276337+wulfland@users.noreply.github.com>
Date:   Thu Sep 17 13:32:05 2020 +0200

```

Normal change

```

❯ ~source/repos/basta/UnterDerHaube git ⌂ main ..... 13:32:23 ⓘ
❯ git cat-file -p 124b36
tree e52701bc3e95de2279598fb8a41e5f5bae9bf1bf
parent d7b9f5205e34558ede629149e27772c2101a6f4f
author Michael Kaufmann <5276337+wulfland@users.noreply.github.com> 1600342325 +0200
committer Michael Kaufmann <5276337+wulfland@users.noreply.github.com> 1600342325 +0200

```

Normal change

```

❯ ~source/repos/basta/UnterDerHaube git ⌂ main ..... 13:32:40 ⓘ
❯

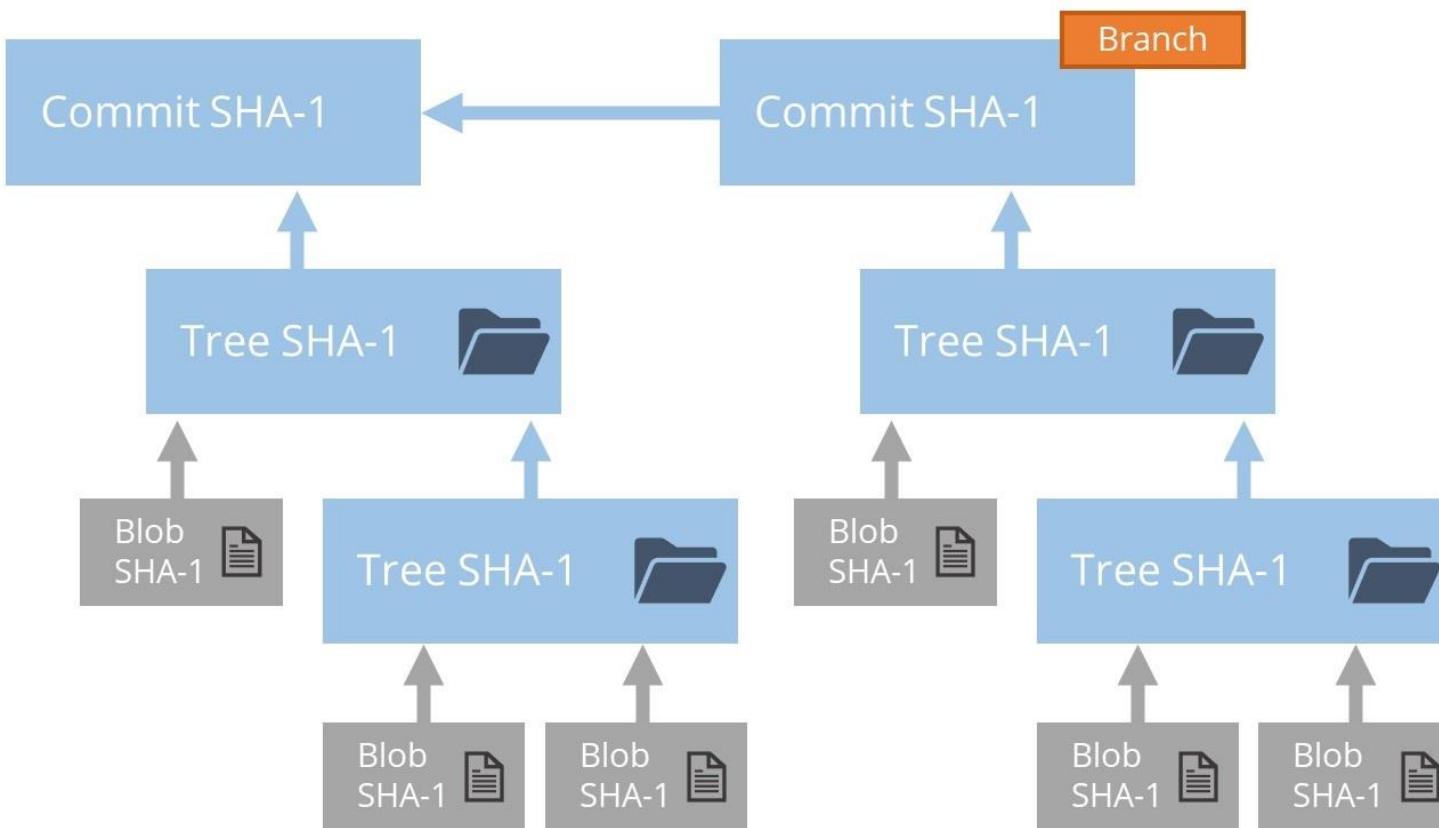
```

```

Every 2.0s: tree -a
.
  .git
    COMMIT_EDITMSG
    HEAD
    MERGE_RR
    config
    description
    hooks
      applypatch-msg.sample
      commit-msg.sample
      fsmonitor-watchman.sample
      post-update.sample
      pre-applypatch.sample
      pre-commit.sample
      pre-merge-commit.sample
      pre-push.sample
      pre-rebase.sample
      pre-receive.sample
      prepare-commit-msg.sample
      update.sample
    index
    info
      exclude
    logs
      HEAD
      refs
        heads
          main
    objects
      12
        4b36b0d18dbe356e30442105bf7529ebc3e971
      2b
        0df5dc9f2e17d4fd0945a2c6bf851d64aa8030
      40
        9d1da9e721d43e6e42c5fee167e8ba36d5133a
      45
        90a439fb3308d87d6d8f8d9b34f55e9fa6a11a
      7d
        dc5a21ba0210931c432b5589102a04c2fd17cf
      d7
        b9f5205e34558ede629149e27772c2101a6f4f
      d9
        587c473053800b9b074a952db01942800760

```

Was passiert bei einem Commit?



- `$ git add`
 - `$ git hash-object`
 - `$ git update-index`
- `$ git commit`
 - `$ git write-tree`
 - `$ git commit-tree`
 - `$ git update-ref`



Demo: Parent Commit



```
✗ ..UnterDerHaube (zsh)
❯ git cat-file -p 2b0df5dc9f2e17d4fd0945a2c6bf851d64aa8030
Name: Dr. Emmet "Doc" Brown
Original time: 1985
Change
❯ git diff HEAD^!
modified: txt/Dox.txt

@ txt/Dox.txt:1 @
Name: Dr. Emmet "Doc" Brown
Original time: 1985
Change

❯
```

Git
speichert
immer die
komplette
Datei!

```
..UnterDerHaube (zsh)
❯ git count-objects
9 objects, 36 kilobytes
❯ git repack -ad
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (8/8), done.
Total 8 (delta 1), reused 8 (delta 1), pack-reused 0
❯ git count-objects
1 objects, 4 kilobytes
❯ git count-objects -Hv
count: 1
size: 4.00 KiB
in-pack: 8
packs: 1
size-pack: 1.85 KiB
prune-packable: 0
garbage: 0
size-garbage: 0 bytes
❯
```

Every 2.0s: tree -a -I hooks Michaels-MacBook-Pro.local: Thu Sep 17 13:50:09 2020

```
.
+-- .git
    |-- COMMIT_EDITMSG
    |-- HEAD
    |-- MERGE_RR
    |-- config
    |-- description
    |-- index
    |-- info
    |   |-- exclude
    |   |-- refs
    |-- logs
    |   |-- HEAD
    |   |-- refs
    |       |-- heads
    |           |-- main
    |-- objects
    |   |-- 7d
    |       |-- dc5a21ba0210931c432b5589102a04c2fd17cf
    |   |-- info
    |       |-- packs
    |   |-- pack
    |       |-- pack-69f258c9d00a8f7ff10d88ae75ba56bd0ebf99cf.idx
    |       |-- pack-69f258c9d00a8f7ff10d88ae75ba56bd0ebf99cf.pack
    |-- refs
    |   |-- heads
    |       |-- main
    |   |-- tags
    |-- rr-cache
    |-- txt
        |-- Dox.txt

14 directories, 16 files
```

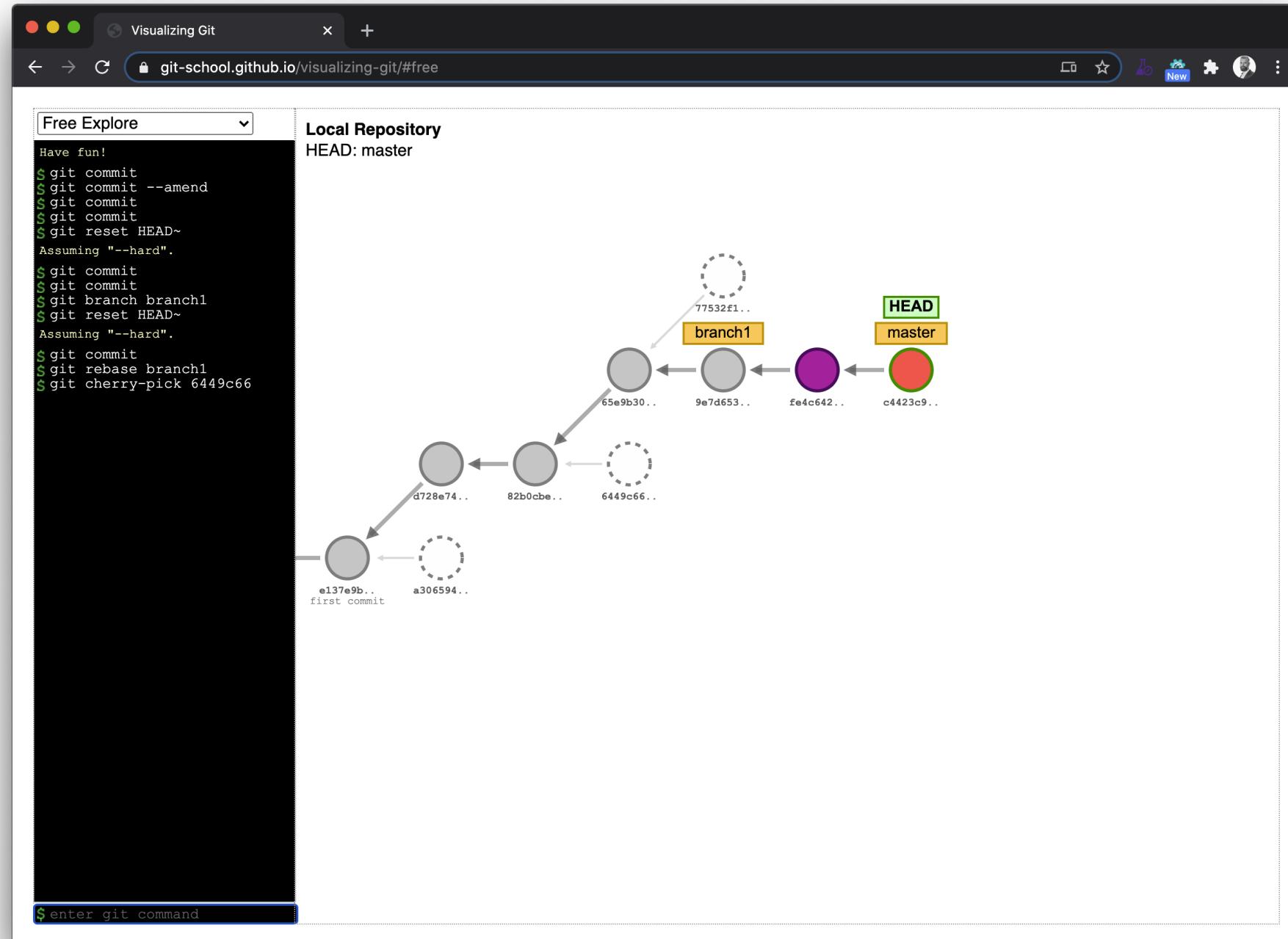


Demo: Patches



```
..UnterDerHaube (zsh)
❯ git format-patch HEAD~2..HEAD
0001-Hallo-Basta2.patch
0002-Hallo-Patch.patch
❯ git reset HEAD~2 --hard
HEAD is now at 139fc85 Hallo Basta
❯ git apply 0001-Hallo-Basta2.patch
❯ git commit -am "patched change"
[main f1a7a86] patched change
 1 file changed, 1 insertion(+)
❯ git am 0002-Hallo-Patch.patch
Applying: Hallo Patch
❯
```

Git
überträgt
immer
Patches!



Zeitreisen in Git



Demo: Rebase Demo



A Clean History

»I want clean history, but that really means (a) clean and (b) history.

People can (and probably should) rebase their *private* trees (their own work). That's a *cleanup*. But never other peoples code. That's a *destroy history*.« – Linus Torvalds





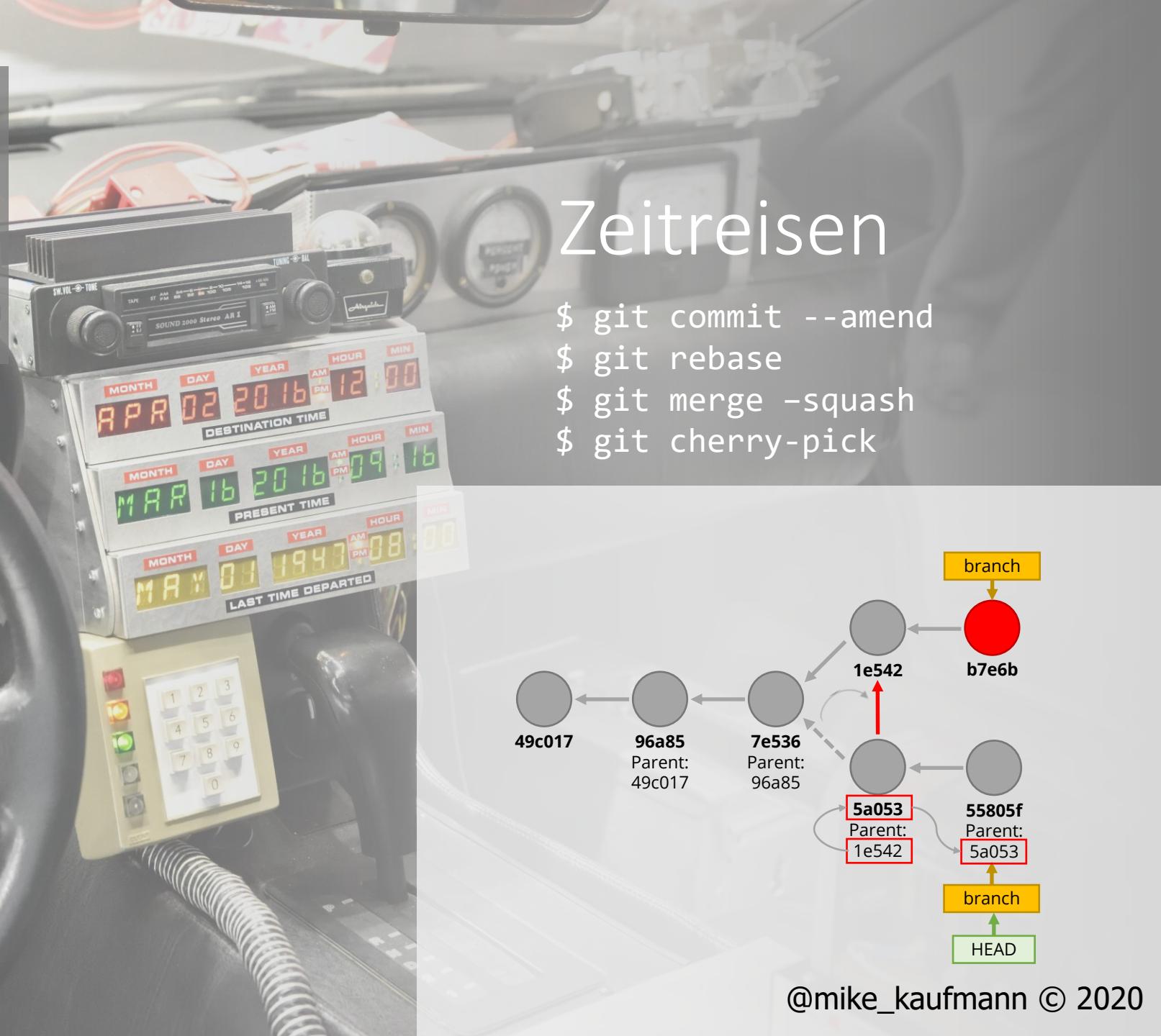
@mike_kaufmann © 2020

Keep your own history readable

» Some people do this by just working things out in their head first, and not making mistakes. But that's very rare, and for the rest of us, we use git rebase etc while we work on our problems. « - Linus Torvalds.

Oberste Direktive:

Verändern Sie niemals die Zeitlinie von anderen Personen! Ändern Sie niemals Ihre eigene Zeitlinie, wenn Sie Bereits Teil der Historie ist.

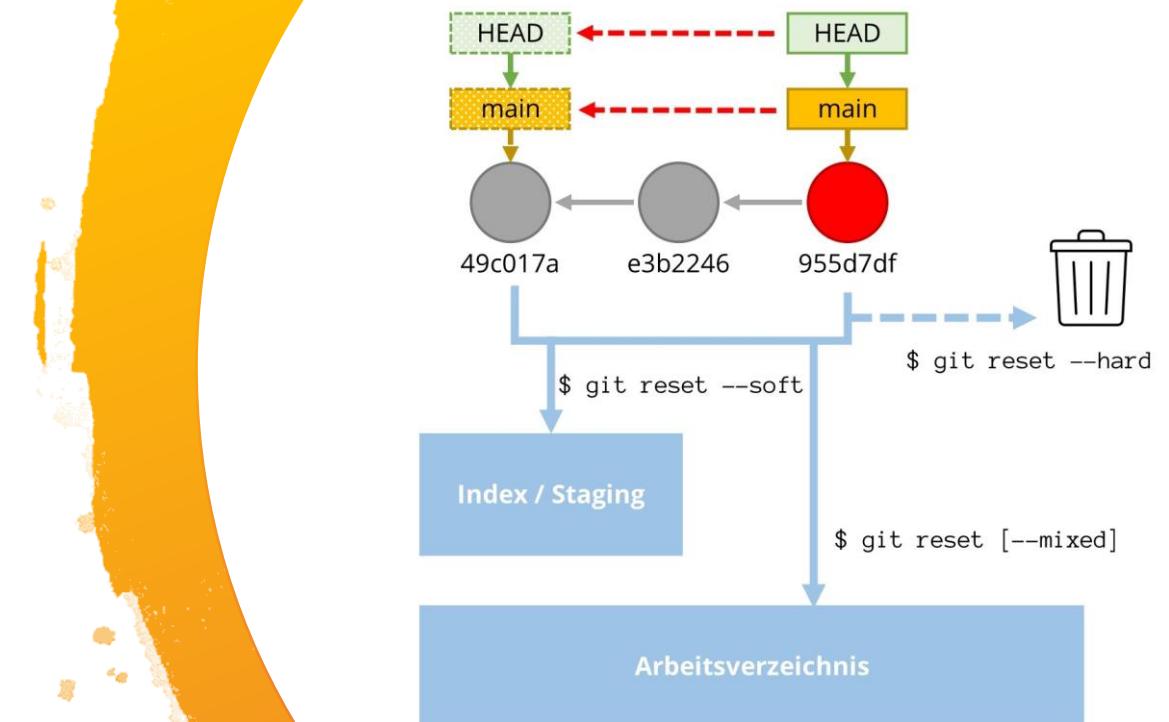




Demo:
Navigation für
Zeitreisen

Zeitreisen mit reset

```
$ git reset REF  
  [--hard |  
   --soft |  
   --mixed]
```



```
..UnterDerHaube (zsh)
❯ ~s/repos/basta/UnterDerHaube git ⌂ main ?2 18:27:37 ⓘ
> git reset --hard 'HEAD@{3}'
HEAD is now at 6126639 Hallo Patch
❯ ~s/repos/basta/UnterDerHaube git ⌂ main ?2 18:27:51 ⓘ
> git reflog
6126639 (HEAD -> main) HEAD@{0}: reset: moving to HEAD@{3}
7738027 HEAD@{1}: am: Hallo Patch
f1a7a86 HEAD@{2}: commit: patched change
139fc85 HEAD@{3}: reset: moving to HEAD~2
6126639 (HEAD -> main) HEAD@{4}: commit: Hallo Patch
c9bdc02 (branch1) HEAD@{5}: reset: moving to HEAD@{13}
ddfbfe8 HEAD@{6}: am: Add -p demo
560b534 HEAD@{7}: commit: Patch Demo
32f9299 HEAD@{8}: reset: moving to HEAD~2
```

Reflog

- Default sind 30 Tage!

```
reflogExpireUnreachable = 30
reflogExpire = 90
pruneExpire = 2.weeks.ago
```

Garbage Collection

Git ruft intern regelmäßig gc bei folgenden Befehlen auf:

- \$ git pull
- \$ git merge
- \$ git rebase
- \$ git commit

Intern ruft gc viele andere Befehle auf, wie zum Beispiel

- \$ git prune,
- \$ git repack oder
- \$ git rerere.

**DON'T
PANIC**

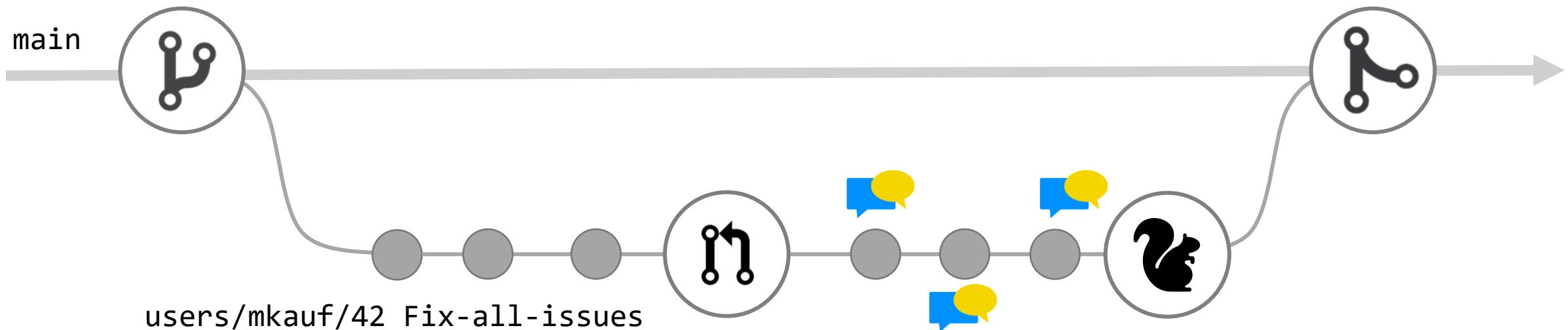
The Hitchhiker's Guide to the Galaxy



Hitchhikers Guide to Time Travel

- Reise mit leichtem Gepäck
(Änderungen in den stash)
- Manipuliere immer nur deine Zeitlinie
- Force-Push immer nur auf „private“ Branches
- Verwende das Reflog als Backup

Beispiel Workflow



Private Branches (users/* private/* etc.)

Zusammenarbeit mit Pull-Requests

Nur der Autor **ändert** seinen Branch (suggestions)

Haupt-Branch **main** ist mit Branch-Protection for Force-Push geschützt

```
$ git push origin +42_Fix-all-issues  
$ git push -force-with-lease origin 42_Fix-all-issues
```



GIT(1)

man git

Git Manual

⌘⌘3

GIT(1)

NAME

git - the stupid content tracker

SYNOPSIS

```
git [--version] [--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>]
     [--super-prefix=<path>]
     <command> [<args>]
```

DESCRIPTION

Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

See **gittutorial(7)** to get started, then see **giteveryday(7)** for a useful minimum set of commands. The **Git User's Manual[1]** has a more in-depth introduction.

After you mastered the basic concepts, you can come back to this page to learn what commands Git offers. You can learn more about individual Git commands with "git help command". **gitcli(7)** manual page gives you an overview of the command-line command syntax.

A formatted and hyperlinked copy of the latest Git documentation can be viewed at <https://git.github.io/htmldocs/git.html> or <https://git-scm.com/docs>.





Vielen Dank

Git verstehen - Unter der Haube der Quellcodeverwaltung...