

GIT Version Control Software

Kevin Noisternig
TINF19B3
DHBW Karlsruhe

Geschichte

Grundlagen

Befehle

Branching

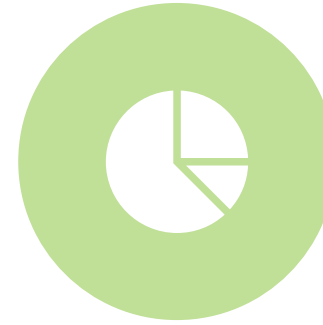
Vergleich



Erscheinungsjahr
2005



Initiator
Linus Torvalds



meistgenutzte
Versionskontroll-
software

- kostenlos
- Open-Source
- einfaches Ändern, nachvollziehen und revidieren von verschiedenen Versionen
- dezentrale Versionsverwaltung
- offline Arbeiten möglich

Setup

```
git config --global user.name  
„[username]“
```

Angezeigter Name in der
Versionshistorie

```
git config --global user.email  
“[valid-email]“
```

Angezeigte Email-Adresse in der
Versionshistorie

```
git config --global color.ui  
auto
```

Färbung der Kommandozeile für
einfacheres Lesen

Initialisieren

`git init`

Initialisieren eines lokalen git Repository

`git clone [url]`

Klonen eines entfernten Repository

Befehle

Grundlagen

Geschichte

Branching

Vergleich

Stage und Commit

`git status`

Zeigt den Status des derzeitigen Repository an

`git add [file]`

Fügt ausgewählte Datei der nächsten Stage hinzu

`git commit -m „[message]“`

Stage wird im Repository übernommen

`git reset [file]`

Änderungen werden verworfen und der letzte Stage wird wiederhergestellt

`git diff`

Zeigt die Unterschiede der Änderungen zum letzten Stage

`git diff --staged`

Zeigt die Änderungen, welche dem Stage hinzugefügt worden sind

Branching

Vergleich

Befehle

Grundlagen

Geschichte

Branch und Merge

<code>git branch</code>	Zeigt alle Branches im derzeitigen Repository an
<code>git branch [branch-name]</code>	Erstellt einen neuen Branch
<code>git checkout [branch-name]</code>	Wechselt zum ausgewählten Branch
<code>git merge [branch]</code>	Fügt ausgewählter Branch in derzeitigen Branch ein
<code>git log</code>	Zeigt alle Commits im derzeitigen Branch an

Teilen und Updaten

`git remote add [alias] [url]`

Einfügen eines entfernten Repository

`git fetch [alias]`

Alle Änderungen von [alias]
runterladen aber nicht integrieren

`git pull [alias] [branch]`

Änderungen herunterladen und direkt
integrieren

`git push [alias] [branch]`

Lokale Änderungen auf entferntes
Repository hochladen

Dateiänderung 1

```
$ git add .
```

```
$ git commit -m „Dateiänderung_1“
```

```
$ git branch feature
```

```
$ git checkout feature
```

Dateiänderung 2

```
$ git add .
```

```
$ git commit -m „Dateiänderung_2“
```

Dateiänderung 3

```
$ git add .
```

```
$ git commit -m „Dateiänderung_3“
```

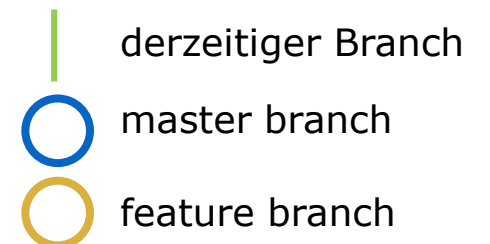
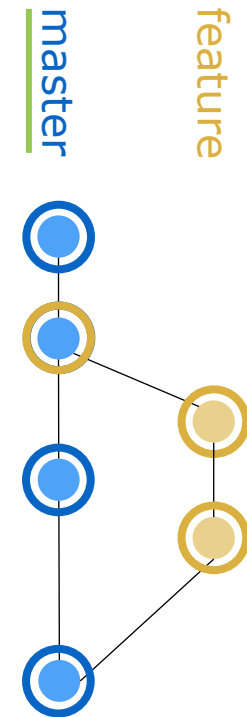
```
$ git checkout master
```

Dateiänderung 4

```
$ git add .
```

```
$ git commit -m „Dateiänderung_4“
```

```
$ git merge feature
```





	GitHub	GitLab
Kosten	Privatnutzung bis 3 Entwickler kostenlos	Privatnutzung kostenlos
Continuous Integration	keine (nur über Drittanbieter möglich)	kostenlos
registrierte Nutzer	50 Millionen	30 Millionen
Unternehmenszugehörigkeit	Microsoft (seit 2018)	Unabhängig
Hosten auf eigenem Server	Kostenpflichtig	Kostenlos
IDE	Minimalistisch	Integrierte Entwicklungsumgebung

Praktische Übung

Geschichte

Grundlagen

Befehle

Branching

Vergleich