

Git



git

Was ist Git?

Git:

- **Git** ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die durch Linus Torvalds initiiert wurde
- Der Name **Git** bedeutet in der britischen Umgangssprache Blödmann.
- “I’m an egotistical bastard, and I name all my projects after myself. First ‘Linux’, now ‘Git’.” - *Linus Torvalds*
- **Git** ist ein verteiltes Versionsverwaltungssystem, das sich von typischen Versionskontrollsystemen unterscheidet

Benutzung

Repository anlegen:

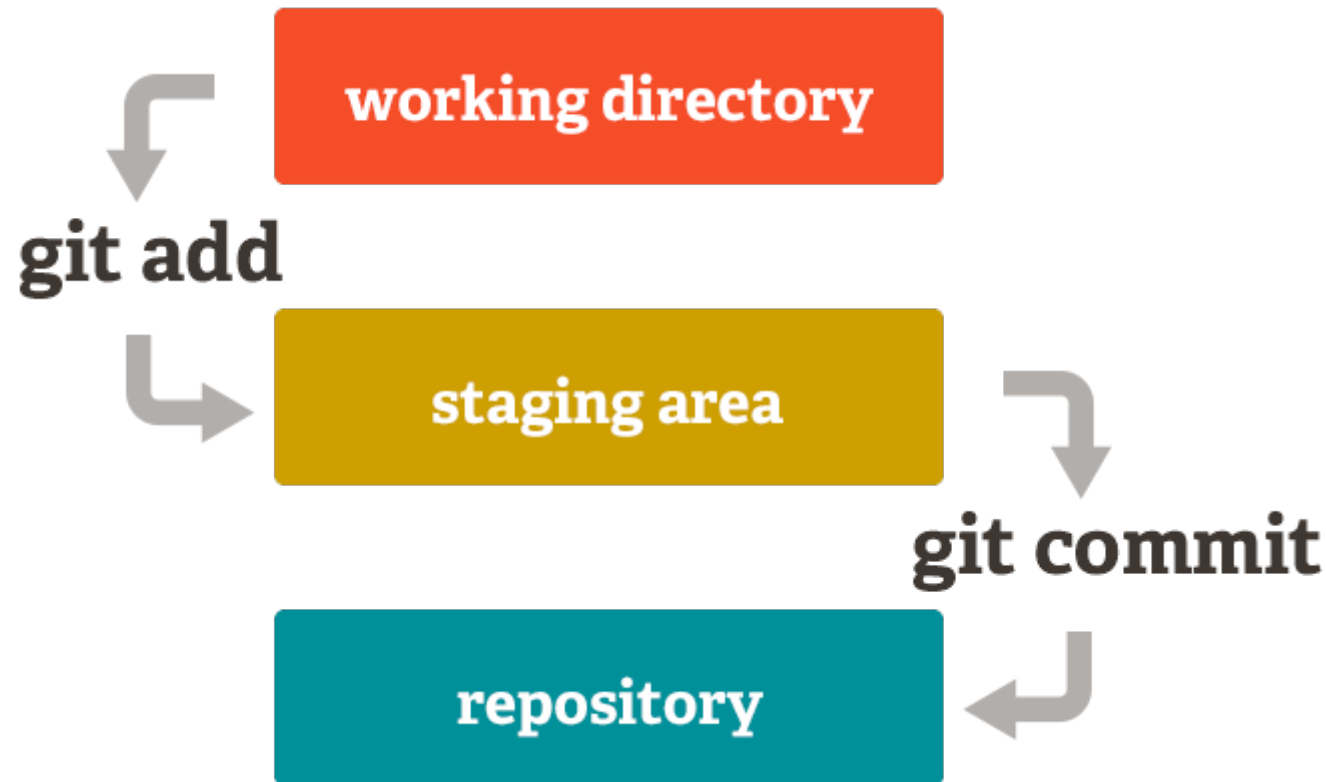
- ***git init*** – legt ein neues lokales git-Repository an
- Dies erstellt ein Ordner Namens ***.git***, mit folgenden Unterverzeichnissen:
HEAD branches config description hooks info objects refs
- Im ***HEAD*** steht der aktuelle branch (Standard: master)
- ***Branches*** sind mehrere Versionszweige
- Repository auschecken:
git clone /pfad/zum/repository
git clone benutzername@host:/pfad/zum/repository

Benutzung

Repository bearbeiten:

- Eine Repository besteht aus 3 Instanzen:
Working dir → ***Staging area (Index)*** → ***Repository (Head)***
- **Working dir**: Arbeitskopie, die echte Dateien enthält
Staging area: Zwischenstufe
Head: Zeigt auf den letzten commit
- ***git add <dateiname>*** – schlägt Änderungen vor (zum Index hinzufügen)
- ***git commit -m „Commit-Nachricht“*** - bestätigt die Änderungen, die sich jetzt im Head befinden, aber noch nicht im **entfernten** Repository

Was ist Git?



Benutzung

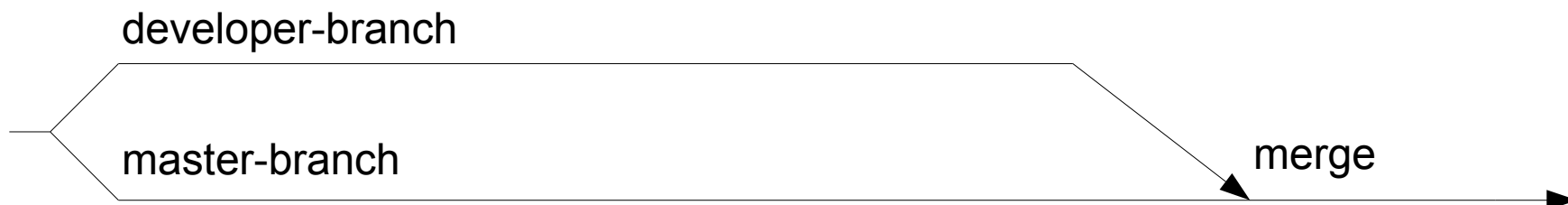
Änderungen hochladen:

- ***git push origin master*** – lädt die Änderungen hoch auf den master-branch
- **origin** – ein „remote“ (Sagt woher die Repository herkommt, nützlich wenn man die Änderungen auf unterschiedliche Server hochladen möchte)
- ***git remote add <name> <server>*** - Fügt ein anderes remote hinzu
- Als Server können sie gerne *https://gitlab.spline.de* benutzen
- Dazu müssen Sie zuerst einen Account auf *https://accounts.spline.de* erstellen

Benutzung

Branches:

- Um unterschiedliche Versionen voneinander getrennt zu halten benutzt man **Branches**
- Das Standard-branch heißt **master**
- Für Entwicklung ist es am besten auf einem anderen Branch zu arbeiten und anschließend mit dem **master-branch** zu mergen



Benutzung

Branches:

- ***git checkout <branch>*** - wechselt zu einem existierenden Branch
- ***git checkout -b <branch>*** - erstellt ein Branch und wechselt zu diesem
- ***git branch -d <branch>*** - löscht ein Branch
- ***git push origin <branch>*** - lädt den Branch hoch
- ***git diff <branch1> <branch2>*** - Zeigt die Unterschiede zwischen 2 Branches
- ***git merge <branch>*** - Führt ein Branch mit dem Aktuellen zusammen

Benutzung

Aktualisieren und Versionen:

- ***git pull*** – aktualisiert deine Arbeitskopie (lädt die Änderungen herunter und führt sie mit deinem Stand zusammen)
- ***git tag <Version> <Commit-ID>*** - Setzt ein Releasetag für ein bestimmtes Commit
- ***git log*** – listet alle Commits-IDs
- ***git checkout -- <dateiname>*** - Setzt die Änderungen zurück, die nicht commited sind
- ***git fetch origin*** – lädt die Repository runter ohne zu mergen
- ***git reset --hard origin/master*** – Löscht alle deine lokale Änderungen und wird auf den Stand der Repository zurückgesetzt