

# Eine Einführung in Git

Paul Nykiel

2. Mai 2020

# 1 Einleitung

# 2 Konzept

# 3 Nutzung

# 4 GitLab

# 5 Git bei uns

# 6 Abschluss

# 7 Praxis

**Einleitung**

Konzept

Nutzung

GitLab

Git bei uns

Abschluss

Praxis

# Einleitung

# Wofür ein Versionsverwaltungssystem?

Situation: Mehrere Leute arbeiten über längere Zeit an einer Codebase

# Wofür ein Versionsverwaltungssystem?

Situation: Mehrere Leute arbeiten über längere Zeit an einer Codebase

Probleme:

# Wofür ein Versionsverwaltungssystem?

Situation: Mehrere Leute arbeiten über längere Zeit an einer Codebase

Probleme:

- Datei Austausch zwischen Nutzer

# Wofür ein Versionsverwaltungssystem?

Situation: Mehrere Leute arbeiten über längere Zeit an einer Codebase

Probleme:

- Datei Austausch zwischen Nutzer
- Aber: nicht sofort, erst wenn fertig

# Wofür ein Versionsverwaltungssystem?

Situation: Mehrere Leute arbeiten über längere Zeit an einer Codebase

Probleme:

- Datei Austausch zwischen Nutzer
- Aber: nicht sofort, erst wenn fertig
- Alte Codestände sollten getestet werden können



# Wofür ein Versionsverwaltungssystem?

Situation: Mehrere Leute arbeiten über längere Zeit an einer Codebase

Probleme:

- Datei Austausch zwischen Nutzer
- Aber: nicht sofort, erst wenn fertig
- Alte Codestände sollten getestet werden können

Einleitung

Konzept

Nutzung

GitLab

Git bei uns

Abschluss

Praxis

# Was noch?

- Authentisierung: von wem ist der Code

Einleitung

Konzept

Nutzung

GitLab

Git bei uns

Abschluss

Praxis

# Was noch?

- Authentisierung: von wem ist der Code
- Kein permanenter Internetzugriff

# Was noch?

- Authentisierung: von wem ist der Code
- Kein permanenter Internetzugriff
- Einfache Nutzung

Einleitung

Konzept

Nutzung

GitLab

Git bei uns

Abschluss

Praxis

# Was noch?

- Authentisierung: von wem ist der Code
- Kein permanenter Internetzugriff
- Einfache Nutzung
- Schnell

# Was noch?

- Authentisierung: von wem ist der Code
- Kein permanenter Internetzugriff
- Einfache Nutzung
- Schnell
- Sicher

Einleitung

Konzept

Nutzung

GitLab

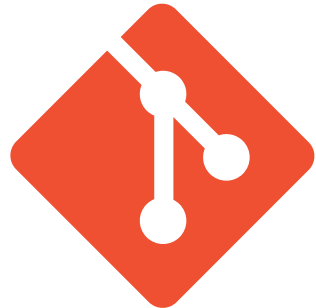
Git bei uns

Abschluss

Praxis

- Freie Software zur Versionsverwaltung

# Git



Einleitung

Konzept

Nutzung

GitLab

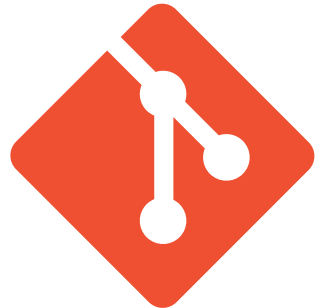
Git bei uns

Abschluss

Praxis

- Freie Software zur Versionsverwaltung
- Dezentral

# Git





## Einleitung

Konzept

Nutzung

GitLab

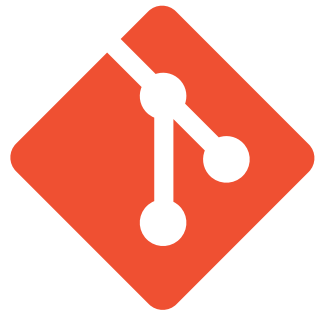
Git bei uns

Abschluss

Praxis

- Freie Software zur Versionsverwaltung
- Dezentral
- Wurde 2005 von Linus Torvals für Linux initiiert

# Git



- Freie Software zur Versionsverwaltung
- Dezentral
- Wurde 2005 von Linus Torvals für Linux initiiert
- Defakto Standard

# Git



# Konzept

- Repository: Eine zusammenhängende Codebase/Projekt

Einleitung

Konzept

Nutzung

GitLab

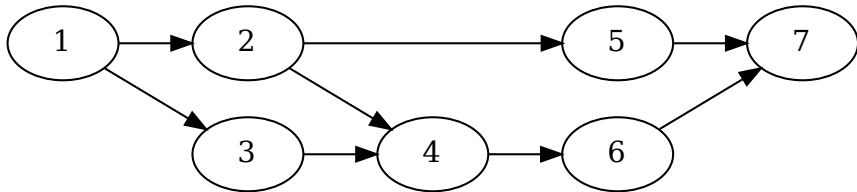
Git bei uns

Abschluss

Praxis

- Repository: Eine zusammenhängende Codebase/Projekt
- Ein Repository ist ein (azyklischer) gerichteter Graph aus Commits

- Repository: Eine zusammenhängende Codebase/Projekt
- Ein Repository ist ein (azyklischer) gerichteter Graph aus Commits



# Commit

- Code-Stand

# Commit

- Code-Stand
- Eindeutige  
Bezeichnung



# Commit

- Code-Stand
- Eindeutige  
Bezeichnung
- Autor

# Commit

- Code-Stand
- Eindeutige  
Bezeichnung
- Autor
- Datum

- Code-Stand
- Eindeutige  
Bezeichnung
- Autor
- Datum
- Nachricht

```
commit feeecb67fe1fa0490a2b836d5ba35da5812a3d27
```

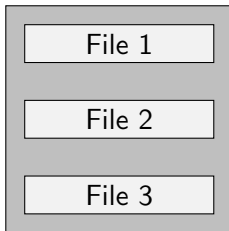
```
Author: Paul Nykiel
```

```
Date:   Mon Apr 20 22:55:41 2020 +0200
```

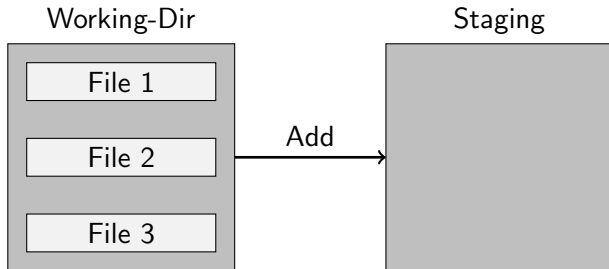
```
Added ADTF type header
```

# Einen Commit anlegen

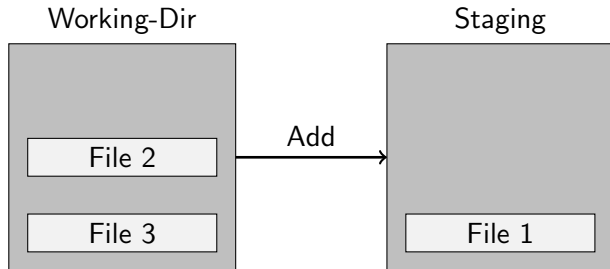
Working-Dir



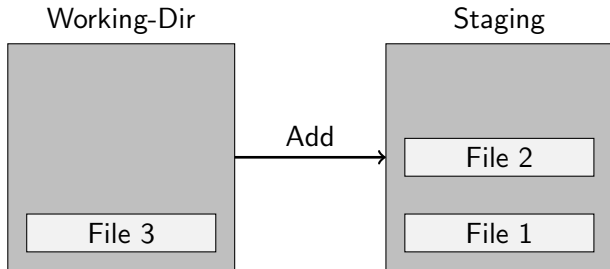
# Einen Commit anlegen



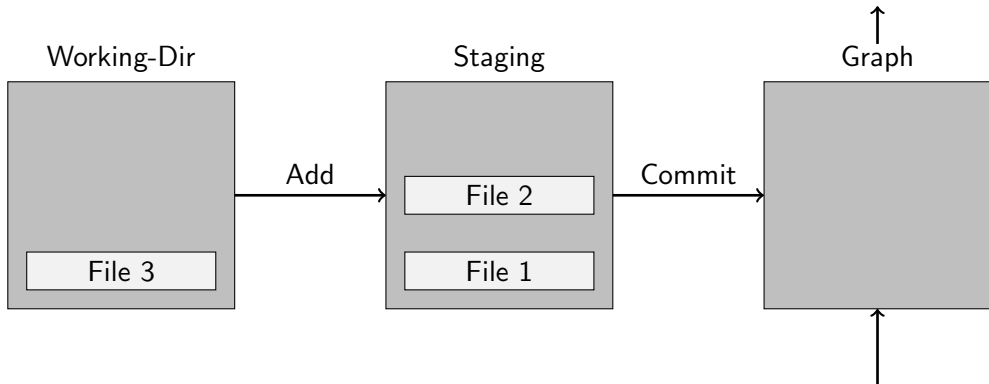
# Einen Commit anlegen



# Einen Commit anlegen

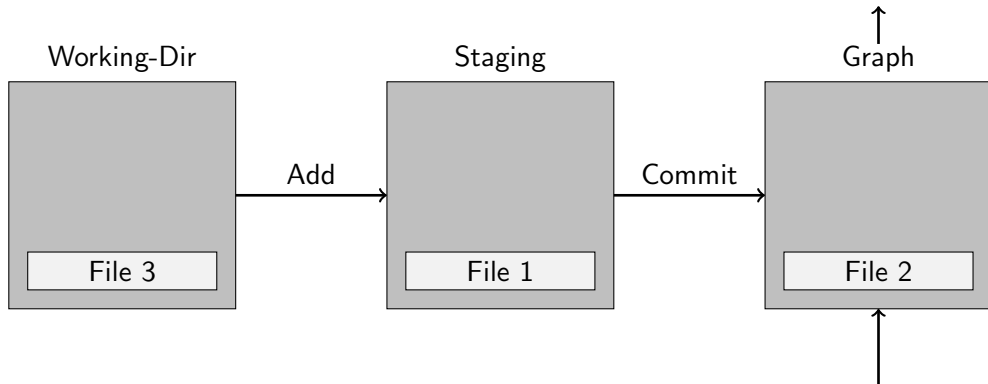


# Einen Commit anlegen



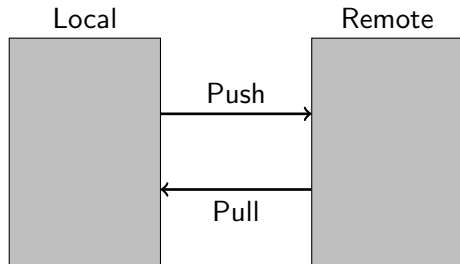


# Einen Commit anlegen



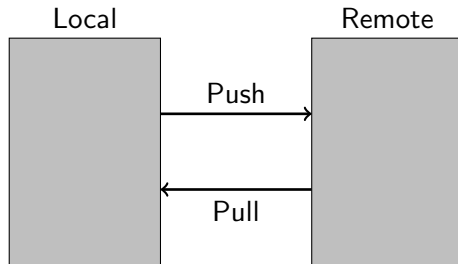
- Bis jetzt alles lokal

# Remotes



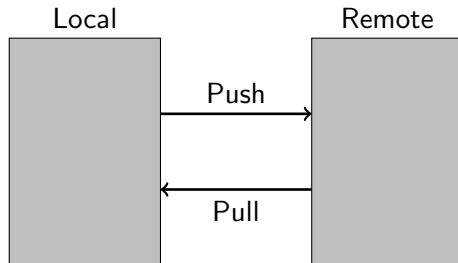
- Bis jetzt alles lokal
- Codebase auf anderem Host:  
„Remote“

# Remotes



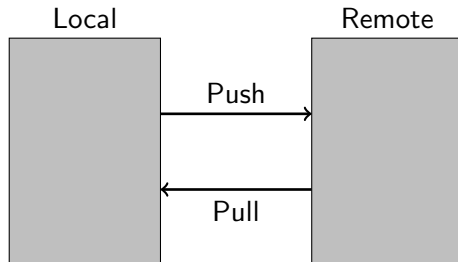
- Bis jetzt alles lokal
- Codebase auf anderem Host: „Remote“
- Operationen: Commits von Remote kopieren, Commits zu Remote kopieren

# Remotes



- Bis jetzt alles lokal
- Codebase auf anderem Host: „Remote“
- Operationen: Commits von Remote kopieren, Commits zu Remote kopieren
- Oftmals ein zentraler Server

# Remotes



# Branches

Einleitung

**Konzept**

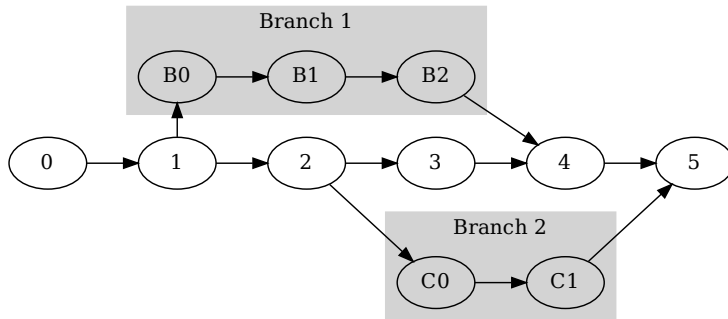
Nutzung

GitLab

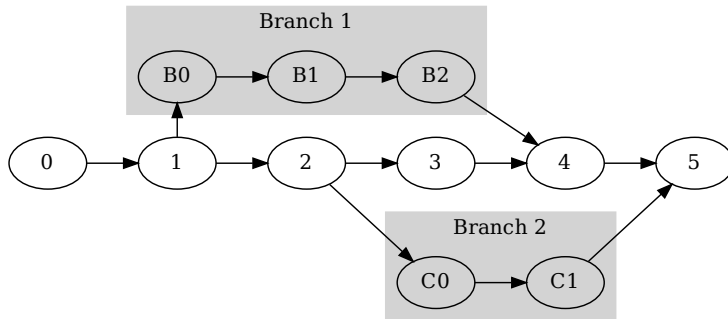
Git bei uns

Abschluss

Praxis



# Branches



Ein Branch ist eine Liste von Commits

# Nutzung



- Manipulation des Zustandes über Git (<https://git-scm.com/>)

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`
  - `commit`

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`
  - `commit`
  - `push`

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`
  - `commit`
  - `push`
  - `pull`

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`
  - `commit`
  - `push`
  - `pull`
  - `checkout`



- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`
  - `commit`
  - `push`
  - `pull`
  - `checkout`
  - `status`

- Manipulation des Zustandes über Git (<https://git-scm.com/>)
- Nutzung: `git command arguments`
- Häufige Befehle:
  - `add`
  - `commit`
  - `push`
  - `pull`
  - `checkout`
  - `status`
  - `log`

add

```
git add FILES...
```

add

```
git add FILES...
```

```
git add -A
```

add

```
git add FILES...
```

```
git add -A
```

```
git add -u
```

# Commit

```
git commit -m "message"
```

# Push

```
git push REMOTE BRANCH
```

# Pull

```
git pull REMOTE BRANCH
```



# Checkout

`git checkout IDENTIFIER`  
mit Identifier: Branch Name, Commit-Hash, ...

# Checkout

`git checkout IDENTIFIER`  
mit Identifier: Branch Name, Commit-Hash, ...

`git checkout -b NEW_BRANCH`

# Information

```
git status
```

# Information

```
git status
```

```
git log
```

# GitLab

# Git bei uns

# Abschluss

# Praxis