

Git Intro

für Einsteiger

Was ist ein
Version
Control
System

Features

Savepoints

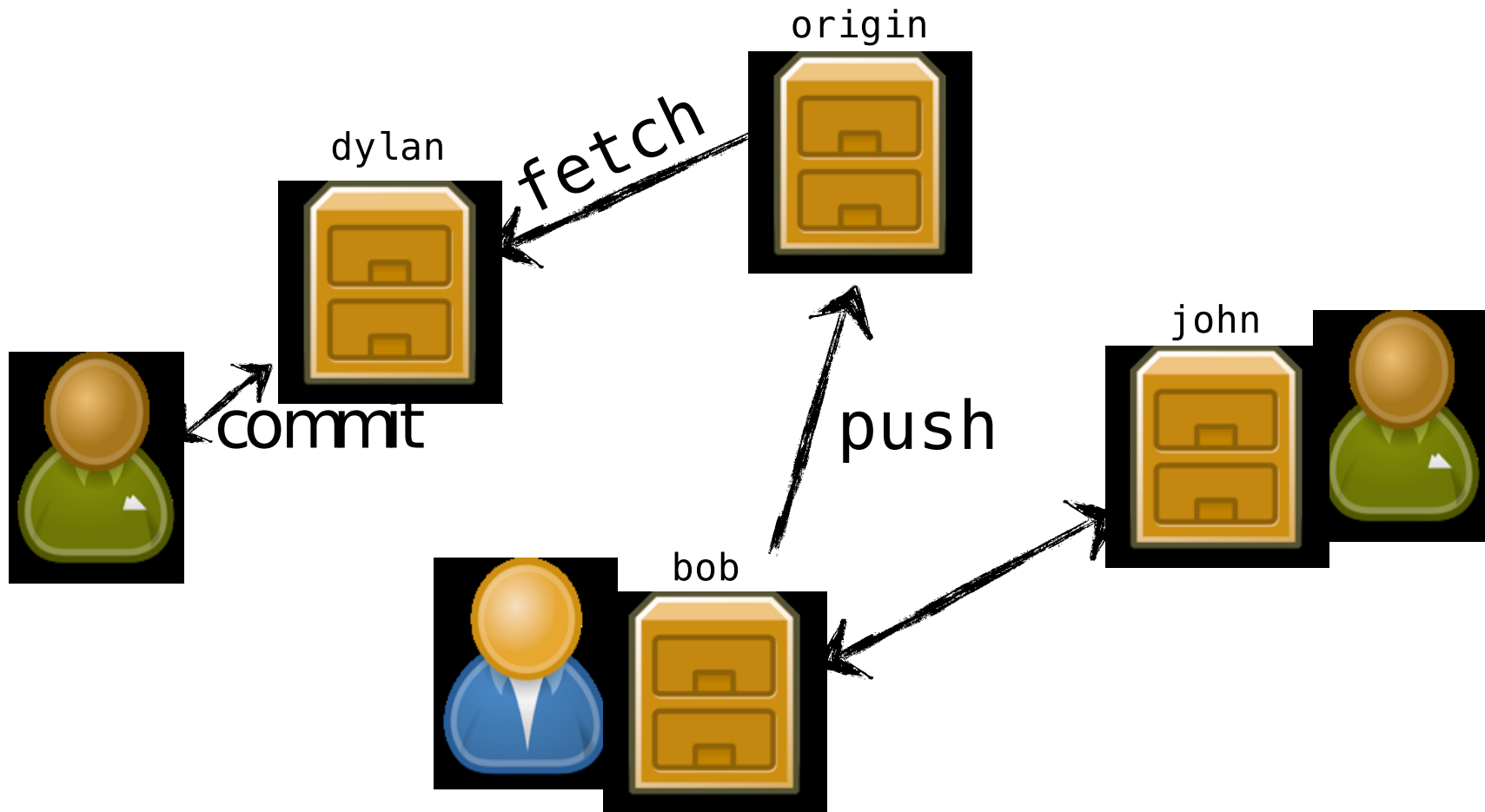
Co-Op

Freies Experimentieren

zentral ^{vs} Verteilt

Zentral

Verteilt



Basics

git: das Tutorial

Setup und Konfiguration

```
git config --global user.name "icke"
```

```
git config --global user.email icke@kiez.de
```


git-Sektionen



A diagram showing two vertical rectangular boxes side-by-side. The left box is orange and contains the text 'working directory'. The right box is blue and contains the text 'repository'.

working
directory

repository

anfangen

init

```
git init
```

```
# erstellt ein neues Repo
```

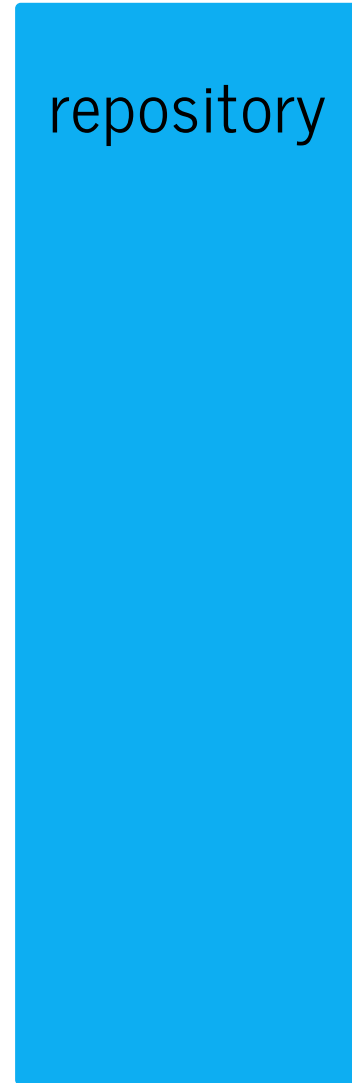
arbeiten

staging area

working
directory

staging
area

repository



arbeiten

add, commit

```
git add
```

```
# working directory → staging area
```

```
git commit [--all] [-m 'Nachricht']
```

```
# staging area → repo
```

commit early, commit often

arbeiten

amend

```
git commit --amend
```

```
# ergänzt den letzten Commit
```

angucken

status, log, diff

```
git status # gut lesbare Status-Anzeige
```

```
git log [--graph] # Commit-History
```

```
git diff # Änderungsliste
```

nimm weg!

cp, mv, rm, .gitignore, reset

```
git rm [--cached]
```

```
# Löscht Dateien [nur im Repo]
```

```
git reset --hard HEAD
```

```
# Zurücksetzen auf den letzten Commit
```

Cheat Sheet

```
git config --global user.name "icke"
```

```
git config --global user.email icke@kiez.de
```

```
git init # erstellt ein neues Repo
```

```
git add # working directory → staging area
```

```
git commit [--all] [--amend] [-m 'Nachricht']  
# staging area → repo
```

```
git status # gut lesbare Status-Anzeige
```

```
git log [--graph] # Commit-History
```

```
git diff # Änderungsliste
```

```
git rm [--cached] # Löscht Dateien [nur im Repo]
```

```
git reset --hard HEAD # Zurücksetzen auf den letzten Commit
```


Branches

um sich dran aufzuhängen

Demo

Cheat Sheet

```
git branch # Liste der Branches
```

```
git branch <name> # Erstelle Branch <name>
```

```
git checkout <name> # Wechsle zu Branch <name>
```

```
git checkout -b <name> # Erstelle und wechsle zu Branch <name>
```

```
git branch -d <name> # Lösche Branch <name>
```

```
git merge <name> # Merged Branch <name> in den aktuellen Branch
```

```
git tag <name> # Erstellt einen Tag
```

```
git tag -a <name> -m 'Nachricht' # Erstellt einen kommentierten Tag
```

Remotes

... andere arbeiten lassen

Demo

Bonus-Feature

stash

stas
h

working
directory

staging
area

repository

```
git stash
```

Aktuelle Änderungen werden zwischengelager

Cheat Sheet

```
git clone <adresse> [<name>] # kopiert Repo von <adresse>
```

```
git remote add origin <adresse> # fügt Remote zu bestehendem Repo hinzu
```

```
git pull [--rebase] # holt Änderungen vom Remote [ohne Merge]
```

```
git push # schiebt Änderungen zum Remote
```

```
git stash # Aktuelle Änderungen werden zwischengelagert
```

```
git stash pop # gespeicherte Änderungen wieder einspielen
```

pssst, bonus-material