# Git Workshop - Let's git our shit together!

## Installation und Accounterstellung

- siehe README des Repositories
- www.github.com/friep/git-our-shit-together

## Partner up!



• Mona Lovalace Octocat



• Grace Hopper Octocat

#### Warum Git?

 $\bullet \quad Master arbeit. docx$ 



Figure 1: Help

- $\bullet \ \ Masterarbeit\_v1.docx$
- $\bullet \quad Master arbeit\_FINAL. docx$
- $\bullet \ \ Master arbeit \_FINAL\_Tims Kommentare. docx$
- $\bullet \ \ Master arbeit \_FINAL\_FINAL.docx$

#### Version Control to the Rescue!

• Beispiel: diese Präsentation.

## Daten runterladen - Fork und Clone

# Fork und Clone @boxbg-blue text-black rounded @box[bg-blue text-black rounded](Fork#"A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project." (Source) Hands On 1 - Fork und Clone Mona @ol • https://github.com/friep/git-our-shit-together/: Fork (oben rechts) • https://github.com/{USERNAME}/git-our-shit-together öffnet sich • unter Settings->Collaborators Grace hinzufügen @olend Hands On 1 - Fork und Clone Mona & Grace @01 • Gitkraken Clone Repo -> Clone with URL • Kopierten Link unter URL eintragen @olend Oh! Oh! - Nicht-Gitkraken Cloning into 'git-our-shit-together'... git@github.com: Permission denied (publickey). fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.

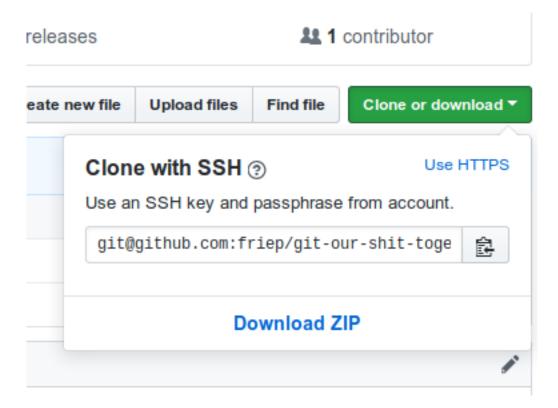


Figure 2: Git clone

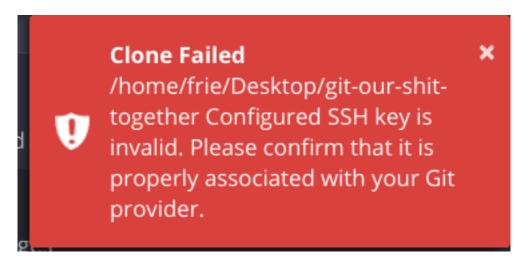


Figure 3: Git clone error

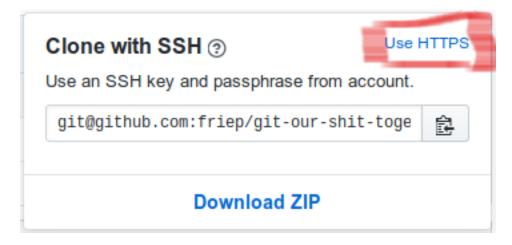


Figure 4: Git clone

#### SSH

#### @ul

- Download:
  - prinzipell jede\*r @fa [lock-open] über https -> @fa [github]: @fa [check]
  - ssh: vorherige Einrichtung ntowendig -> @fa [github]: @fa [question]
- Upload: nur authentifizierte Personen @fa [lock]
- -> @fa [github]: @fa [question]

@ulend

#### Authentification - Passwort

- bei jedem Push GitHub Passwort eingeben
- beachte: clone https://...

#### Authentification - SSH

- public key, private key cryptography (siehe z.B. Youtube)
- nur einmal einrichten ->
- clone ssh://...

#### Hands On 1.1: Gitkraken mit Github verbinden

#### @ol

- Gitkraken Profil (rechts oben)
- Preferences->Authentification->GitHub
- connect to GitHub
- Generate SSH key and add to GitHub

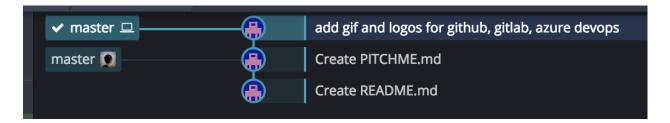


Figure 5: Git commits

@olend

# Daten speichern - Add und Commit

Commit history

Go back in time!

#### Hands On 2 - Go back in time

@ol

- reset master to this commit
- spiele mit: hard, mixed, soft
- fast forward master to origin/master (oberster commit)

@olend

#### Commit

- Commit hält Veränderungen gegenüber dem vorherigen Commit fest
  - Änderungen von Dateien
  - Neuerstellung von Dateien
  - Löschung von Dateien
  - Umbenennung von Dateien
- ein Commit kann mehrere Änderungen beinhalten

Adding und Staging Area

(Source: https://git-scm.com/about/staging-area)



Figure 6: Time Machine

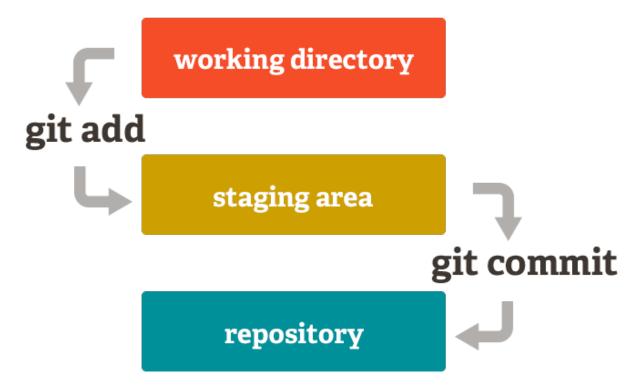


Figure 7: Staging Area

#### Hands On 3 - einen Commit machen

Grace + Mona

@ol

- Change stuff!
- GIT ADD von den "Unstaged Files" Dateien GIT ADDen, die man in Git "speichern" möchte.
- (halbwegs) aussagekräftige Commit Message schreiben
- GIT COMMIT

@olend

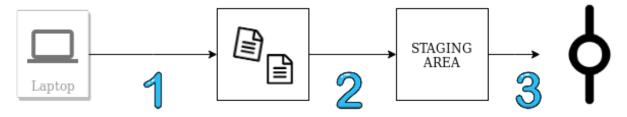


Figure 8: Lokal



Git quizzed!

Git quizzed!

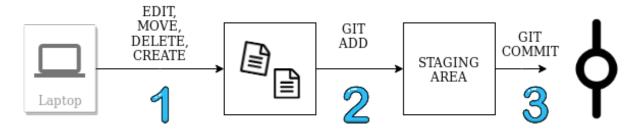


Figure 9: Lokal

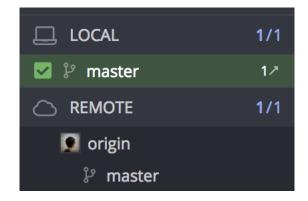


Figure 10: Gitkraken Lokal Remote

# Daten syncen - Push und Pull

## Git Hosting

• Die Cloud! z.B.

@fa [gitlab] @fa [github]

#### Git Lokal und Git Remote

 $\dots$  what?

**Lokal**: dein PC **Remote**: in der Cloud (GitHub, GitLab, ...)

## Sync: Git Pull und Git Push

- Git Pull: neue Commits von GitHub downloaden
- Git Push: lokal erstellte Commits nach GitHub hochladen

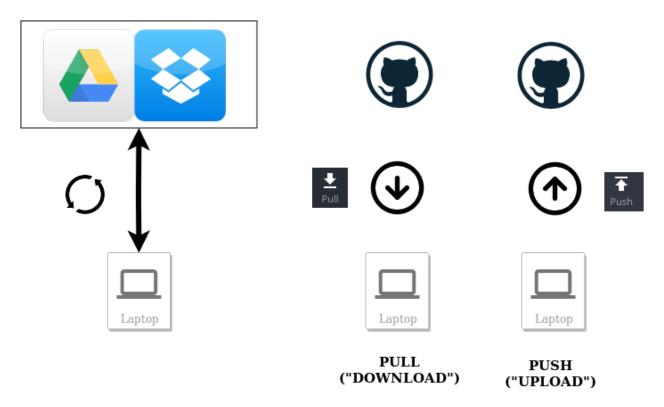


Figure 11: Push Pull

Hands On 4 - Pull und	Push
@ol	
<ul><li>Grace: Push</li><li>Mona: Pull</li><li>Mona: Push</li><li>Grace: Pull</li></ul>	
@olend	
Git quizzed!	
Git quizzed!	

Sync: Git Pull und Git Push



Figure 12: Push the button

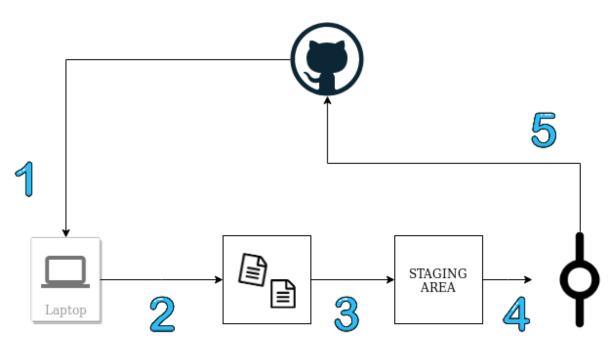


Figure 13: Push Pull

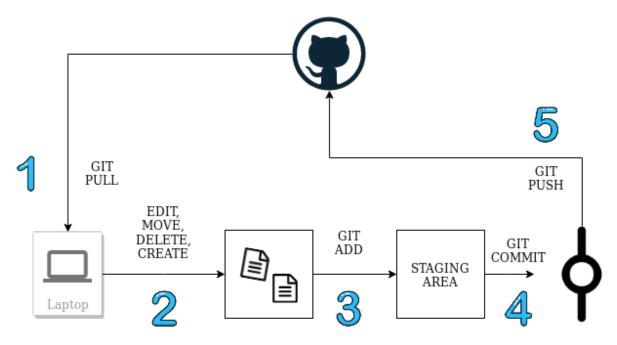


Figure 14: Push Pull

When things go wrong	
@ol	
- oft committen!	sht ist, alles (halbwegs) gut sichern und neu clonen
@olend	
Git stash	
@boxbg-blue text-black ro	unded
Git stash	
-> put it away for now!	

When things go wrong...

git errors Q

Alle Bilder Videos News Shopping Mehr Einstellungen Tools

Ungefähr 44.400.000 Ergebnisse (0,37 Sekunden)

#### Git Tutorial: 10 Common Git Problems and How to Fix Them ...

https://www.codementor.io > Community > Git ▼ Diese Seite übersetzen

21.10.2014 - So far we covered quite a lot of ground on how to fix common errors when working with glt . Most of them have easy enough solutions, however ...

# Git happens! 6 Common Git mistakes and how to fix them | GitLab

https://about.gitlab.com/2018/08/08/git-happens/ ▼ Diese Seite übersetzen

08.08.2018 - If you're brand new to Git, you can learn how to start using Git on the ... After a good few hours of coding, it's easy for a spelling error to sneak ...

## Oh, shit, git!

https://ohshitgit.com/ ▼ Diese Seite übersetzen

Git is hard: screwing up is easy, and figuring out how to fix your mistakes is ... Oh shit, I did something terribly wrong, please tell me git has a magic time machinel?! ... but I've come to these steps through trial and error and lots of swearing and ...

# How to colorify git errors, warnings and fatal messages? - Stack ...

https://stackoverflow.com/.../how-to-colorify-git-errors-warnings... ▼ Diese Seite übersetzen

4 Antworten

21.05.2018 - With Git 2.18 (Q2 2018), you now have a better documentation of the settings to colorize push errors/hints. See commit 79f62e7 (21 Apr 2018) ...

GIT Push Errors - Duplicate Request

18. Juli 2016

How do I fix "remote: fatal error in commit\_refs" errors trying to ...

20. Mai 2016

how to fix GIT error: object file is empty?

11. Sept. 2012

How to deal with Git "Could not read" error

2. Okt. 2009

Weltere Ergebnisse von stackoverflow.com

# GitHub - pkg/errors: Simple error handling primitives

https://github.com/pkg/errors ▼ Diese Seite übersetzen

Simple error handling primitives. Contribute to pkg/errors development by creating an account on GitHub.

## Dealing with non-fast-forward errors - User Documentation - GitHub Help

https://help.github.com/.../dealing-with-non-fast-forward-errors/ ▼ Diese Seite übersetzen

Sometimes, Git can't make your change to a remote repository without losing commits. When this happens, your push is refused. If another person has pushed to ...

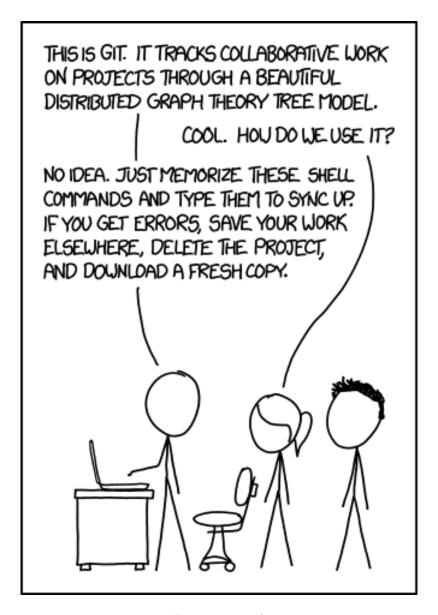


Figure 16: xccd



Figure 17: stash

# Git stash bei merge conflicts

@ol

- git stash
- git pull
- apply stash
- solve merge conflicts
- (delete stash)

@olend

Hands	on 5	): IVI	lerge	conf	licts	

# mit GitHub arbeiten

## Issues

@ul

 $\bullet\,$ issues: Todos / Bugs / Ideen

<ul><li>jeder issue hat eine Nummer</li><li>#issueno in commit message verknüpft commit mit issue</li></ul>	
@ulend	
Hands On 4: Issue	
@ol	
<ul> <li>Mona: Issue erstellen: "Grace's LieblingsGIF fehlt"</li> <li>Grace: füge der Präsentation eine neue Folie hinzu mit deinem Lieblingsgif (g</li> <li>Grace: add + commit. verlinke issue Nummer in der commit message (#issu</li> <li>Grace: push</li> <li>Mona: Issue neu laden (STRG+R)</li> </ul>	
@olend	
Branches	
[picture of complicated gitkraken with a lot of branches]	
Branches	
@boxbg-blue text-black rounded	
Branches	
@boxbg-blue text-black rounded	
Why branches?	
@ul	
<ul> <li>"master" branch frei von unfertigem Code halten</li> <li>unabhängige Entwicklung von Code ("feature branches")</li> <li>Experimente</li> </ul>	
@ulend	



Figure 18: Come on

#### **Branches Workflow**

@ol

- Branch erstellen
- normal weiterarbeiten (pull-commit-push cycles)
- (optional: merge andere branches in deinen branch um Updates zu bekommen)
- merge Branch in master branch

@olend

#### Merging branches

- Rechtsclick auf branch name / master
- hängt davon ab, wer "weiter vorne" ist (?)
  - -wenn neue commits auf master: merge master into #1-add-branch-slides -> branch wird geupdatet
  - wenn neue commits auf branch: merge #1-add-branch-slides into master -> master wird geupdatet

#### **Branches Fazit**

@ul

- besonders nützlich bei Kollaboration
- Entwicklung von Packages
- Relevanz für Datenprojekte (?) -test text

@ulend

Das wars.

gerne den Tag über fragen!