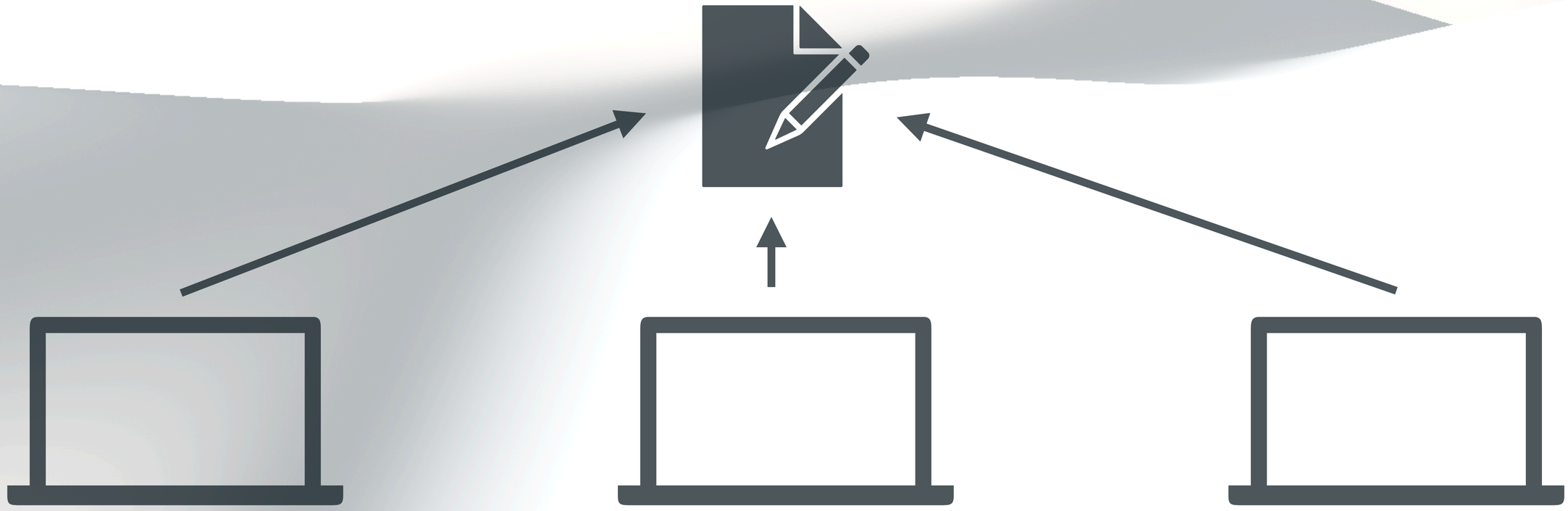
Versionsverwaltung (Git)

Patrick Oster B. Eng.
CEO & Co-Founder of muf:fin
Professional Trader & Investor
Data-Engineer & -Analyst
Vocational- & Work-Educator

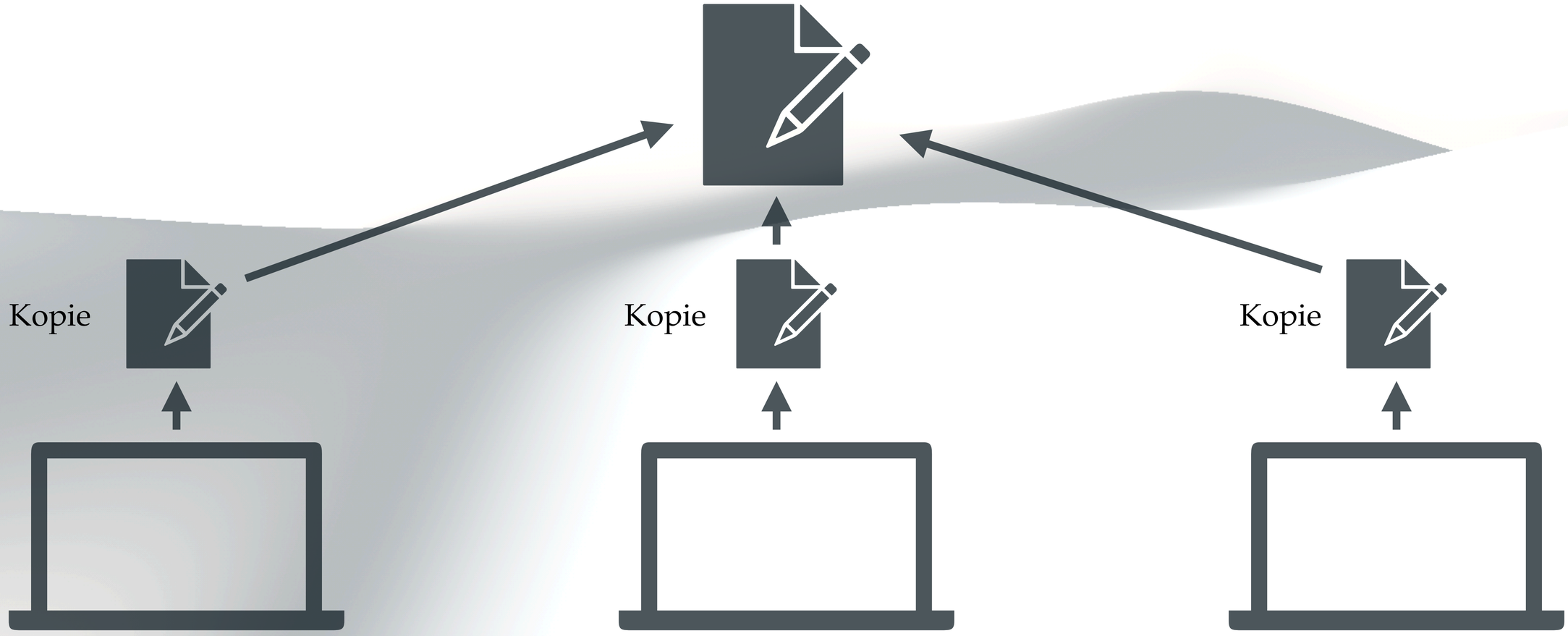
Versionsverwaltung …

Mal angenommen, wir arbeiten allesamt zur gleichen Zeit am selben Dokument … welche Probleme könnten sich daraus ergeben?

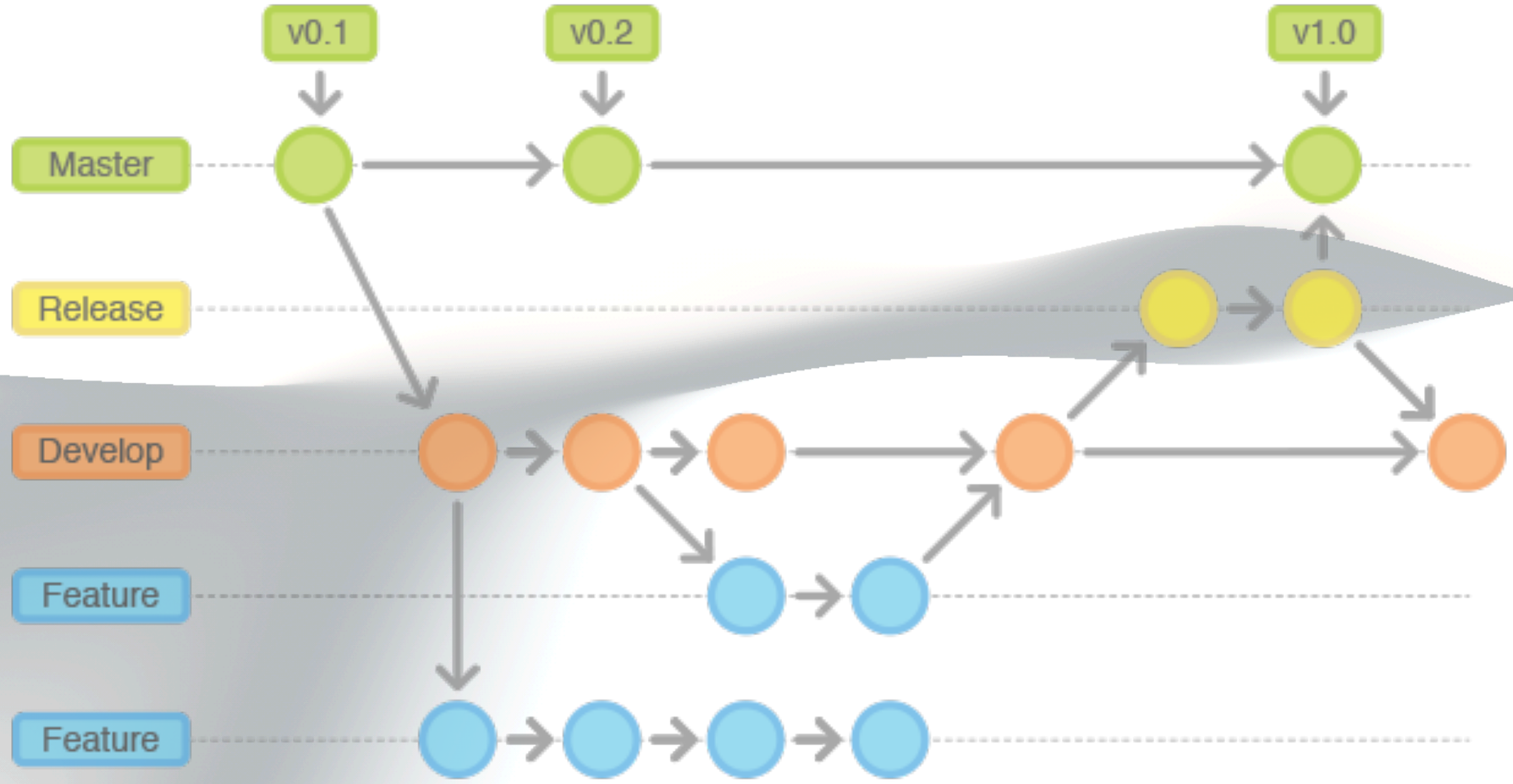


Versionsverwaltung …

Was wäre aber, wenn sich nun jeder Einzelne von uns eine Kopie des Dokuments erstellt?

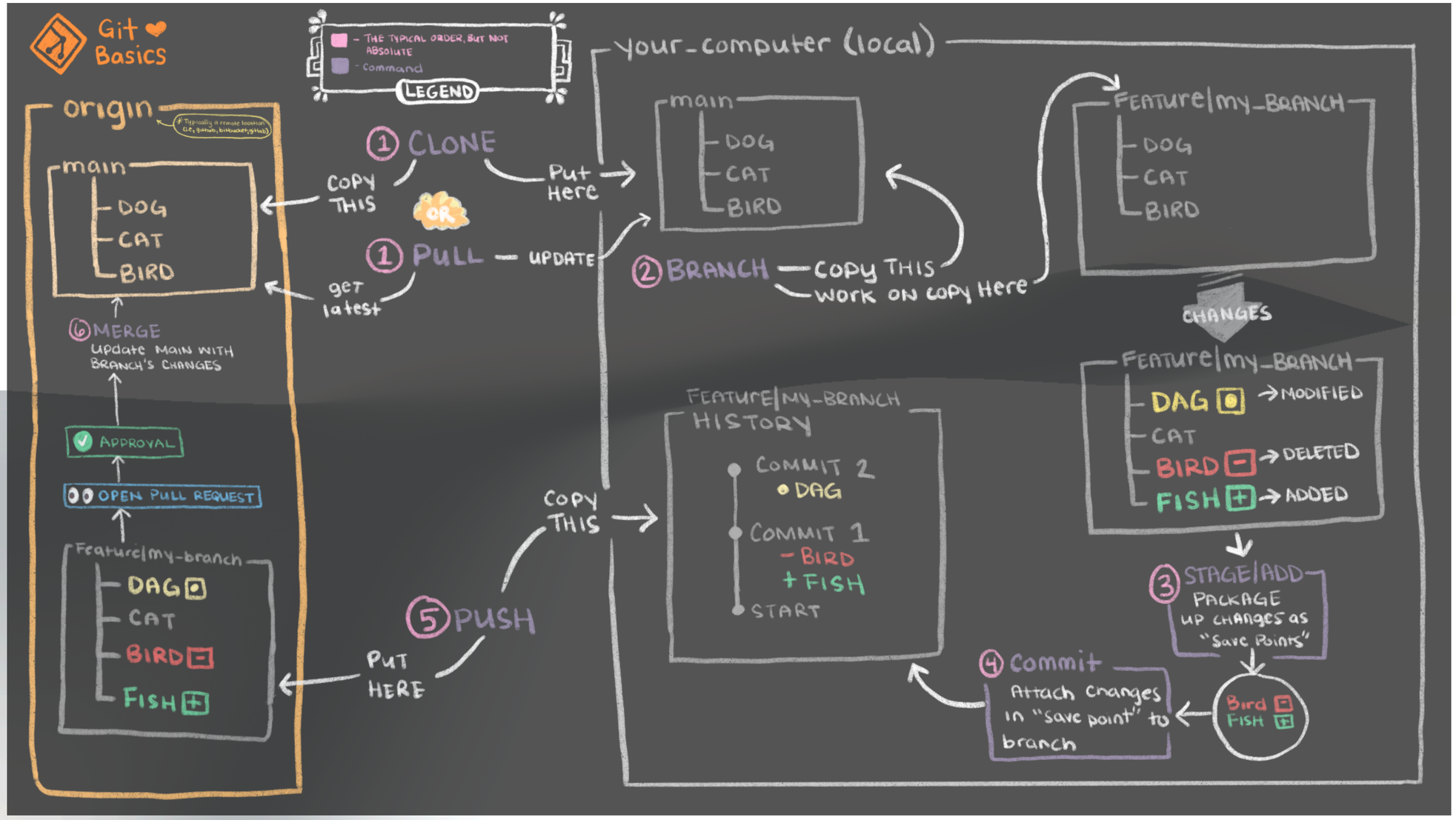


Verteilte Versionsverwaltung (Git) …



Quelle: https://seibert.group/blog/2014/03/31/git-workflows-der-gitflow-workflow-teil-1/

Workflow …


Git ❤ Basics
- THE TYPICAL ORDER, BUT NOT ABSOLUTE
- command
LEGEND
origin
Typically a remote location (i.e. github, bitbucket, github)
main
DOG
CAT
BIRD
1 CLONE
COPY THIS
Put Here
OR
1 PULL
get latest
UPDATE
your_computer (local)
main
DOG
CAT
BIRD
2 BRANCH
COPY THIS
work on copy here
feature/my_branch
DOG
CAT
BIRD
CHANGES
feature/my_branch
DAG → MODIFIED
CAT
BIRD → DELETED
FISH → ADDED
3 STAGE/ADD
PACKAGE UP CHANGES as "Save Points"
Bird
FISH
4 Commit
Attach changes in "save point" to branch
feature/my-branch HISTORY
COMMIT 2
• DAG
COMMIT 1
- BIRD
+ FISH
START
COPY THIS
5 PUSH
PUT HERE
feature/my-branch
DAG
CAT
BIRD
FISH
6 MERGE
Update main with branch's changes
APPROVAL
OPEN PULL REQUEST

Cheat sheet …

Git Cheat Sheet

REBELLABS by ZEROTURNAROUND

Create a Repository

From scratch -- Create a new local repository
```
$ git init [project name]
```

Download from an existing repository
```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed
```
$ git status
```

Show the changes to files not yet staged
```
$ git diff
```

Show the changes to staged files
```
$ git diff --cached
```

Show all staged and unstaged file changes
```
$ git diff HEAD
```

Show the changes between two commit ids
```
$ git diff commit1 commit2
```

List the change dates and authors for a file
```
$ git blame [file]
```

Show the file changes for a commit id and/or file
```
$ git show [commit]:[file]
```

Show full change history
```
$ git log
```

Show change history for file/directory including diffs
```
$ git log -p [file/directory]
```

Working with Branches

List all local branches
```
$ git branch
```

List all branches, local and remote
```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory
```
$ git checkout my_branch
```

Create a new branch called new_branch
```
$ git branch new_branch
```

Delete the branch called my_branch
```
$ git branch -d my_branch
```

Merge branch_a into branch_b
```
$ git checkout branch_b
$ git merge branch_a
```

Tag the current commit
```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit
```
$ git add [file]
```

Stage all changed files, ready for commit
```
$ git add .
```

Commit all staged files to versioned history
```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history
```
$git commit -am "commit message"
```

Unstages file, keeping the file changes
```
$ git reset [file]
```

Revert everything to the last commit
```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)
```
$ git fetch
```

Fetch the latest changes from origin and merge
```
$ git pull
```

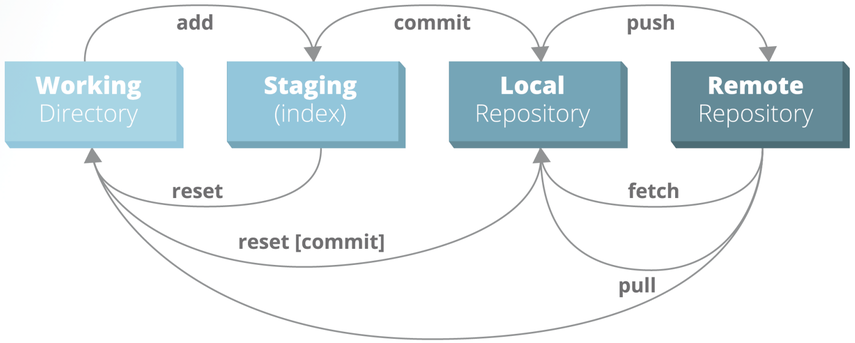
Fetch the latest changes from origin and rebase
```
$ git pull --rebase
```

Push local changes to the origin
```
$ git push
```

Finally!

When in doubt, use git help
```
$ git command --help
```

Or visit https://training.github.com/ for official GitHub training.



Übung für Selbstlernphase …

a) *Machen Sie sich bitte noch näher mit G**it** vertraut! Schauen Sie sich bitte hierzu folgendes Video an und machen Sie sich ggf. ein paar Notizen, sollte Ihnen etwas unklar sein: https://www.youtube.com/watch?v=uGLQF2kUwOA*

b) *Nehmen Sie das Beispiel mit der „Einkaufsliste“ und versuchen Sie das Gelernte anzuwenden. Gehen Sie einfach davon aus, Sie leben in einer Wohngemeinschaft, in der alle ihren Einkaufszettel zu einer einzigen, großen Einkaufsliste zusammenfügen wollen (Stichwort „Branching“).*