Git Cheatsheet

Configure Tooling

`$ git config --global user.name "[name]"`
Sets the name for your commit messages

`$ git config --global user.email "[email]"`
Sets the email for your commit messages

`$ git config --global --list`
View your configuration

`$ git config --global core.editor [editor]`
Sets the editor for Commands such as commit and tag that lets you edit messages

Create/Clone Repositories

`$ git init [directory]`
Creates a new local repository in the specified directory

`$ git clone [url]`
Downloads a repository and its entire version history

Make changes

`$ git status`
Lists all new or modified files to be commited

`$ git add [file]`
Adds the file to the staging area

`$ git rm [file]`
Deletes the file from the working copy and stages the deletion

`$ git reset [file]`
Unstages the file, but preserve its contents

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

`$ git blame [file]`
Who changed what and when in a file

`$ git diff`
Shows file differences not yet staged

`$ git diff --staged`
Shows file differences between staging and the last file version in the repository

`$ git commit -m "[descriptive message]"`
Records changes permanently in version history of the repository

`$ git commit --amend`
Change the latest commit

`$ git revert HEAD`
Reverts the changes of the latest commit

`$ git stash`
Temporarily saves changes from the working copy offsite and reverts the working copy

`$ git stash pop`
Restores the most recently stashed files

Remote Repositories

`$ git remote add [remote] [url]`
Adds a tracked remote repository with name [remote] and URL [url]

`$ git remote -v`
Lists all tracked remote repositories

`$ git fetch [remote]`
Downloads all history from the repository remote

`$ git merge [remote]`
Combines remote branch into current local branch

`$ git push [remote] [branch]`
Uploads all local branch commits to remote repo

`$ git push -u [remote] [branch]`
Additional to upload add upstream reference fro every branch

`$ git push -f`
Push history rewriting commits

`$ git pull [remote]`
Downloads commits and incorporates changes

`$ git pull --rebase`
Replace the current branch with the remote branch and apply local commits on top of it

Review History

`$ git log`
Lists version history for the current branch

`$ git log --oneline`
Condense each commit to a single line

`$ git log -n`
Shows the latest n commits

`$ git show [commit]`
Outputs metadata and content changes of the specified commit

Branches & Tags

`$ git branch`
Lists all local branches in the current repository

`$ git branch [branch-name]`
Creates a new branch

`$ git merge [branch]`
Combines the specified branch's history into the current branch

`$ git branch -d <branch>`
Deletes the specified branch

`$ git tag my-tag`
Tag the current commit

`$ git cherry-pick [commit]`
Apply then changes of a commit which exists on another branch

Navigate Repository

`$ git checkout <branch>`
Switches to the specified branch and updates the working copy

`$ git checkout <commit>`
Switches to the specified commit and updates the working copy

Links

Kursunterlagen

https://github.com/cko/if2017-git

Offizielle Git Dokumentation

https://git-scm.com/doc

Tutorials & Bücher

https://www.atlassian.com/git/tutorials
https://git-scm.com/book/en/v2
http://rogerdudler.github.io/git-guide/