

# Cheat sheet Github cmd's

## Basics:

- Git clone <url> => clone a repository to your machine
- Git pull => get the current version of your branch or the master branch
- Git push => push any changes from your local machine into github.com
- Git status => check for changed Files
- Git add => Adding all or parts of your changed file for a commit
- Git commit -m "test" => Summarizing all your addings into one commit with a caption
- Git push => Pushing it directly into the main

## Working in a team:

- Git pull
- Git checkout -b "test\_branch" => Based on your main/master-branch you create a copy where you develop your feature on
- Git add
- Git commit
- Git push => Now a url is generated, click on it to create a Pull Request
- On github.com you can based on this create a Pull request or a Draft Pull Request
  - If many people are affected by your code changes or if you going to do many commits afterwards do a "Draft" one, so there are not permanently informed by all your upcoming commits
- All further "git push"-cmds directly put your changes on your Pull Request

## Special cmd's:

- Git merge master/main => if your branch is too outdated you can do a merge with master branch
  - Merge master is first merging your changes and then the others
- Git rebase => also a merge with master branch
  - First all other changes are merged an then yours

- Git cherry-pick <commit> => if you need the feature of a colleague in your branch you can cherry-pick it
- Git fetch --all => get all branches of the corresponding repository

## Scenarios:

### Squash all commits of your PR

- Count the commits
- Double check the number with git log on your machine
- On your branch do “git reset HEAD~<number> –soft”
- Git commit -m “Squashin”
- Git push

### Remove unwanted files from your PR

- Git co master -- <file\_path>
- Git commit -m “remove”
- Git push

### Undo your latest change

- git reset HEAD~1–hard