

Cheat Sheet für GitHub:

Grundlegende Befehle:

- **git init:** Initialisiert ein neues Git-Repository.
- **git clone <repository_url>:** Klont ein bestehendes Repository in einen neuen Ordner.
- **git add <file>:** Fügt eine Datei zur Staging-Area hinzu.
- **git commit -m "<message>":** Erstellt einen Commit mit einer Nachricht.
- **git push:** Überträgt lokale Änderungen auf den Remote-Repository.
- **git pull:** Aktualisiert lokale Kopie mit den neuesten Änderungen aus dem Remote-Repository.

Branches:

- **git branch:** Zeigt eine Liste der vorhandenen Branches an.
- **git checkout <branch_name>:** Wechselt zu einem anderen Branch.
- **git checkout -b <new_branch_name>:** Erstellt einen neuen Branch und wechselt zu ihm.
- **git merge <branch_name>:** Führt Änderungen aus einem anderen Branch in den aktuellen Branch zusammen.
- **git branch -d <branch_name>:** Löscht einen lokalen Branch.

Remote-Repository:

- **git remote -v:** Zeigt die URLs der Remote-Repositorys an.
- **git remote add <remote_name> <remote_url>:** Fügt ein Remote-Repository hinzu.
- **git remote remove <remote_name>:** Entfernt ein Remote-Repository.
- **git push -u <remote_name> <branch_name>:** Überträgt lokale Änderungen zum Remote-Repository und verknüpft den Branch.

Änderungen rückgängig machen:

- **git reset <file>:** Setzt Änderungen an einer Datei in den letzten Commit zurück.
- **git reset --hard:** Setzt alle Änderungen in der Arbeitskopie und im Index auf den letzten Commit zurück.
- **git revert <commit_hash>:** Erstellt einen neuen Commit, der die Änderungen eines bestimmten Commits rückgängig macht.

Fortgeschrittene Funktionen:

- **git stash:** Versteckt lokale Änderungen, um an anderer Stelle fortzufahren.
- **git cherry-pick <commit_hash>:** Überträgt die Änderungen eines bestimmten Commits auf den aktuellen Branch.
- **git rebase <base_branch>:** Verschiebt die Commit-Historie

eines Branches auf den letzten Stand der Basis.

- **git bisect:** Hilft bei der Suche nach einem fehlerhaften Commit durch binäre Suche.

GitHub-Workflow:

- **Fork:** Eine Kopie eines Repositories in deinem eigenen GitHub-Konto erstellen.
- **Pull Request:** Eine Änderung vorschlagen und um eine Überprüfung durch den Repository-Besitzer bitten.
- **Code Review:** Überprüfung von vorgeschlagenen Änderungen durch Teammitglieder.
- **Merge:** Zusammenführen von genehmigten Änderungen in den Hauptbranch des Repositories.

GitHub Pages:

- **Branch:** Veröffentliche statische Websites direkt aus dem Master-Branch deines Repositories.
- **CNAME:** Konfiguriere eine benutzerdefinierte Domain für deine GitHub Pages.
- **Themes:** Verwende vorgefertigte Themes, um das Erscheinungsbild deiner GitHub Pages anzupassen.
- **Jekyll:** Verwende Jekyll, um dynamische Inhalte in deine GitHub Pages einzubinden.

Dieses Cheat Sheet bietet eine schnelle Übersicht über grundlegende Git- und GitHub-Befehle sowie fortgeschrittene Funktionen und den GitHub-Workflow. Du kannst es als Referenz verwenden, um effektiv mit Git und GitHub zu arbeiten und deine Projekte zu verwalten.