# **GitHub Cheat Sheet**

A comprehensive cheat sheet covering essential GitHub commands, workflows, and features for version control and collaboration.

## **Basic Git Commands**

#### Configuration

CHEAT

git config global user.name "Your Name"	Sets your name for commit messages.
git config global user.email "email@examp le.com"	Sets your email address for commit messages.
git config global core.editor "code wait"	Sets VS Code as the default editor for commit messages (replace codewait with your editor command).
git config list	Lists all git configuration settings.

## Starting a Repository

git init	Initializes a new Git repository in the current directory.
git clone <repository_ur l&gt;</repository_ur 	Clones a repository from a remote URL.

### **Basic Workflow**

git status	Shows the status of the working directory and staging area.
git add <file></file>	Adds a file to the staging area.
git add .	Adds all changes in the current directory to the staging area.
git commit -m "Commit message"	Commits the staged changes with a descriptive message.

## **Branching and Merging**

#### **Branch Management**

git branch

git branch <branch\_name

checkout <branch\_name

checkout -b

<branch\_name

git branch

<branch\_name

git branch

<branch\_name

> git

> git

>

- d

>

- D

>

#### Merging Branches

Lists all local branches. The current branch is marked with an asterisk (*).	git merge <branch_name></branch_name>	Merges the specified branch into the current branch.
Creates a new branch.	git mergetool	Opens a merge tool to resolve conflicts.
Switches to the specified branch.	git loggraph oneline decorate	Visualize the branch and merge history.
	Rebasing	
Creates and switches to a new branch.	git Rek rebase the <branch_na cor<br="">me&gt;</branch_na>	bases the current branch onto e specified branch. Rewrites mmit history.
Deletes a branch (only if it has been merged).	git Abo rebase abort	ort the rebase process.
Forces deletion of a branch (even if it hasn't been merged).	git Con rebase continue	ntinue the rebase process after olving conflicts.

### **Remote Repositories**

#### Connecting to Remotes

git remote add origin <repository_url></repository_url>	Adds a remote repository named 'origin'.
git remote -v	Lists all remote connections.
git remote show origin	Shows information about the remote 'origin'.

### **Pushing and Pulling**

#### **GitHub Specifics**

git push origin <branch_ name&gt;</branch_ 	Pushes the specified branch to the remote 'origin'.	g <s <e g <r< td=""></r<></e </s 
git push -u origin <branch_ name&gt;</branch_ 	Sets up tracking information for the branch so that git pull and git push can be used without specifying the remote and branch.	
git pull origin <branch_ name&gt;</branch_ 	Pulls changes from the remote 'origin' to the specified branch.	
git fetch	Fetches all changes from the remote but does not merge them.	

# git request-pull Generates a request pull <start> <url> message. <end> Clone a repository. gh repo clone Clone a repository. <repo> Requires GitHub CLI (gh ). (gh ).

## **Undoing Changes**

#### Undoing Local Changes

# Undoing Commits

git checkout - - <file></file>	Discards changes in the working directory for a specific file.
git restore <file></file>	Alternative to git checkout <file> for discarding local changes (more modern).</file>
git reset HEAD <file></file>	Removes a file from the staging area.
git clean -n	Dry run to see which files would be removed by git clean.
git clean -f	Removes untracked files from the working directory.

git commit amend -m "New commit message"	Amends the last commit with new changes or message.
git resetsoft HEAD~1	Resets the last commit, keeping the changes in the staging area.
git reset mixed HEAD~1	Resets the last commit, keeping the changes in the working directory.
git resethard HEAD~1	Resets the last commit and discards the changes. Use with CAUTION.
<pre>git revert <commit_hash></commit_hash></pre>	Creates a new commit that reverts the changes from the specified commit. Safe way to undo changes in shared repositories.