The Winner Sheet › Git

Not a Cheat Sheet about Git commands for version control. Use 'git help <command>' for details.

User Name and E-mail

git config --global user.email
Get global default E-mail address for your commits

git config --global user.name
Get global default user name used for your commits

git config --global user.email <your@email.com>
Set the default user email used globally

git config --global user.name "<Your name>"
Set the default user identity used globally

git config user.name "<Your name>"
Set the user name locally. Similarly, set user.email

git config user.email
Get user e-mail used locally. Similarly, check user.name

Initialise or Clone

git init .
Initialise a GIT repository in the current folder

git clone git@github.com:<user>/<repo>.git
Clone a GIT repository via SSH

git clone https://<token>@github.com/<user>/<repo>
Using tokens to clone a GIT repository via HTTPS

Remotes

git remote -v
See tracked remotes

git remote add origin git@github.com:<user>/<repo>.git
Add a remote under alias "origin"

git remote show origin
Inspecting remote

git remote rename origin1 origin2
Rename remote origin1 to new name origin2

Branches and Merging

git branch -a
List all -a branches, both remote and local

git branch <branch>
Create a branch

git checkout <branch>
Switch to the branch

git branch -d <branch>
Delete a branch with option -d

git push <remote> --delete <branch>
Delete a remote branch

Adding Files and Commits

git add <filename>
Add a file to the staged area (for the next commit)

git commit -m "Commit message"
Commit staged files with the message, use -m option

git commit --amend -m "New commit message"
Edit the latest commit message with --amend option
(when commit is published, use "git push <remote> <branch> --force"

Updates and Publishing

git fetch <remote>
Download all changes from remote without including them into HEAD

git pull <remote> <branch>
Download remote changes and integrate into HEAD

git push <remote> <branch>
Publish local changes to remote

History

git status
Check the status

git log
Check the commits log (for brevity use --oneline option

git blame <filename>
Who changed the file and where

Merging and Rebase

git merge <branch>
Merge a branch into the main branch, in result including all changes

git checkout <branch>; git merge main
Merge main into the branch

git mergetool
Resolve merge conflicts

git rebase <branch>
Rebase HEAD into your branch (apply your changes on top the latest updates of other developers)

Differences

git diff
See local changes that can be further added to commit

git diff --staged
See staged changes with --staged option

git diff --base <filename>
Differences against a base (--ours for your changes, or --theirs)

git diff origin/master
See differences between local and remote

git diff <commit_id1> <commit_id2>
See differences between two commits

A Contribution Workflow

git clone https://<token>@github.com/<user>/<repo>
Cloning a remote (forked) repository locally with your token

git remote add upstream https://github.com/<org>/<repo>
Add upstream for your organisation

git checkout -b <branch>
Working on a new branch, created with -b option

git fetch <remote>
Get changes from remote

git add <filename>
Add changed file to the staged area

git commit -m "commit message"
Commit changes

git push -u <remote> <branch>
Use -u for your first push of the branch to remote

In your GitHub fork page press "Pull Request" button
Create a pull request

Undo Changes

git reset --mixed HEAD~
Removing the last commit, --mixed (default) preserves working tree

git reset --hard HEAD
Undo all changes in your local directory (consider stashing)

git reset <commit_id>
Resets commit (removes commits from the current branch and changes commit history), useful to undo changes that are not yet published

git reset --hard <commit_id>
Resets commit to the defined one, discards all later changes

git revert <commit_id>
Reverts commit to the defined commit, adds a new commit and is a safe operation

git checkout <filename>
Revert file to the latest commit (effects the working directory)

git checkout HEAD <filename>
Discard changes in your local file

Stashing

git stash
Save the current state of your directory and the index

git stash list
Show list of your stashes

git stash pop
Removes the changes from your stash and re-applies them

git stash apply
Apply the changes and keep them in your stash