# Git Cheat Sheet

## Create

From existing data
```
cd ~/my_project_dir
git init
git add .
```
From existing repo
```
git clone ~/existing/repo ~/new/repo
git clone you@host.org:dir/project.git
```
default protocol is ssh

## Change

Using your favorite editor / IDE

## Update

Fetch latest changes from origin
```
git fetch
```
this does not merge them

Pull latest changes from origin
```
git pull
```
does a fetch followed by a merge

Apply a patch that someone sent you
```
git am -3 patch.mbox
```
In case of conflict, resolve the conflict and
```
git am --resolve
```

## Browse

Files changed in working directory
```
git status
```
Changes to tracked files
```
git diff
```
Changes between ID1 and ID2
```
git diff <ID1> <ID2>
```
History of changes
```
git log
```
Who changed what and when in a file
```
git blame <file>
```
A commit identified by ID
```
git show <ID>
```
A specific file from a specific ID
```
git diff <ID>:<FILE>
```
Search for patterns
```
git grep <pattern> [path]
```

## Revert

Return to the last committed state
```
git checkout -f | git reset --hard
```
you cannot undo a hard reset

Revert the last commit
```
git revert HEAD
```
Creates a new commit

Revert specific commit
```
git revert $id
```
Creates a new commit

Fix the last commit
```
git commit -a --amend
```
after editing the broken files

Checkout the ID version of a file
```
git checkout <ID> <file>
```

## Commit

Commit all local changes
```
git commit -a
```

## Publish

Prepare a patch for other developers
```
git format-patch origin
```
Push changes to origin
```
git push [origin] [branch]
```
Make a version or milestone
```
git tag <version_name>
```

## Branch

List all branches
```
git branch
```
Switch to the BRANCH branch
```
git checkout <BRANCH>
```
Merge branch B1 into branch B2
```
git checkout <B2>
git merge <B1>
```
Create branch based on HEAD
```
git branch <BRANCH>
```
Create branch based on another
```
git checkout <new> <base>
```
Delete a branch
```
git branch -d <branch>
```

## Useful tips

Get help
```
git help [command]
```
Create empty branch
```
git symbolic-ref HEAD
refs/heads/newbranch
rm .git/index
git clean -fdx
<do work>
git add your files
git commit -m 'Initial commit'
```
Graphical log
```
git log --graph
git log --graph --pretty=oneline --
abbrev-commit
```
Push branch to remote
```
git push <origin> <branch>
```
Delete remote branch and locally
```
git push <origin> :<branch>
git branch -d <branch>
```

## Resolve merge conflicts

View merge conflicts
```
git diff
```
View merge conflicts against base file
```
git diff --base <FILE>
```
View merge conflicts against other changes
```
git diff --theirs <FILE>
```
View merge conflicts against your changes
```
git diff --ours <FILE>
```
After resolving conflicts, merge with
```
git add <CONFLICTING_FILE>
git rebase --continue
```

## Configuration

```
git config [--global]
```
global is stored in ~/.gitconfig

user
```
user.name $name
user.email $email
```
color
```
color.ui auto
```
github
```
github.user $user
github.token $token
```
optimisation
```
pack.threads 0
diff.renamelimit 0
```
windows
```
core.autocrlf true
```

~/.git/
config, refs, HEAD, index, objects

~/.git/objects/
Three objects: blob = SHA-> blob
tree = list of SHA blob
commit = sha -> tree

~/.git/refs/
heads
remotes

do not use on low memory p