

Git basics commands

Git init

Git clone

Or

Git remote add 'alias' 'url'

Server url

`http://server-name.md/repo-name.git`

`https://server-name.md/repo-name.git`

`git://server-name.md/repo-name.git`

File statuses

tracked, untracked, ignored

Tracked file statuses

unmodified, modified, or staged

Git stage

Git stash

.gitignore

<https://github.com/github/gitignore>

(The Linux kernel source repository has 206 .gitignore files.)

Git remove tracked files

```
git rm --cached [-r]
```

git log

--pretty=oneline

--graph

```
git log --oneline --decorate --graph --all
```

git commit

--ammend

Change only the comment will change the commit hash?

git remote

-v

How we can use them?

git tag

simple vs annotated ?

git alias

```
git config --global alias.co checkout  
git config --global alias.st status
```

Git Branching

Create new branch

```
git checkout -b 'branch-name'
```

Delete branch

```
git branch -d 'branch-name'
```

Merge branch

- `git checkout master`
- `git merge other-branch [--no-ff] [--squash]`

Rebase branch

- `git rebase master [-i]`

Some tricks

Change message on last commit

```
git commit --amend
```

Some tricks

commit on different branch

```
# create a new branch from the current state of master  
git branch some-new-branch-name
```

```
# remove the commit from the master branch  
git reset HEAD~ --hard
```

```
# your commit lives in this branch now :)  
git checkout some-new-branch-name
```

Some tricks

commit on master branch

```
# create a new branch from the current state of master  
git branch some-new-branch-name
```

```
# remove the commit from the master branch  
git reset HEAD~ --hard
```

```
# your commit lives in this branch now :)  
git checkout some-new-branch-name
```

Some tricks

committed to the wrong branch

```
# undo the last commit, but leave the changes available  
git reset HEAD~ --soft
```

```
git stash  
# move to the correct branch  
git checkout name-of-the-correct-branch  
git stash pop  
git add .  
git commit -m "your message here"  
# now your changes are on the correct branch
```


Some tricks

committed to the wrong branch V2

```
git checkout name-of-the-correct-branch
```

```
# grab the last commit to master  
git cherry-pick master
```

```
# delete it from master  
git checkout master  
git reset HEAD~ --hard
```