

# Git

## Add tag

```
git tag TAG
```

## Commit a tag

```
git push origin TAG
```

## Commit all tags

```
git push origin --tags
```

Source: <https://stackoverflow.com/questions/18216991/create-a-tag-in-a-github-repository>

## Clone a specific tag

```
git clone --depth 1 --branch <tag_name> <repo_url>
```

## Specify the private SSH-key to use when executing a Git command:

```
ssh-agent bash -c 'ssh-add /home/git/.ssh/key-filename; git $GIT_COMMAND'
```

Example:

```
ssh-agent bash -c 'ssh-add /home/git/.ssh/key-filename; git push'
```

Source:

<https://stackoverflow.com/questions/4565700/how-to-specify-the-private-ssh-key-to-use-when-executing-shell-command-on-git>

## Remove submodule

1. `mv a/submodule a/submodule_tmp`
2. `git submodule deinit -f - a/submodule`
3. `rm -rf .git/modules/a/submodule`
4. `git rm -f a/submodule` # Note: a/submodule (no trailing slash)

**or, if you want to leave it in your working tree and have done step 0**

5. `git rm --cached a/submodule`

3bis `mv a/submodule_tmp a/submodule`

Source: <https://stackoverflow.com/questions/1260748/how-do-i-remove-a-submodule>

## git undo all uncommitted or unsaved changes

```
git checkout .
```

Or:

```
git checkout [some_dir|file.txt]
```

Source: <https://stackoverflow.com/questions/14075581/git-undo-all-uncommitted-or-unsaved-changes>

## How to undo local changes to a specific file

```
git checkout -- <file>
```

Source: <https://stackoverflow.com/questions/31281679/how-to-undo-local-changes-to-a-specific-file>

## Undo all uncommitted or unsaved changes

Source: <https://stackoverflow.com/questions/14075581/git-undo-all-uncommitted-or-unsaved-changes>

## Getting the difference between two repositories

```
git remote add -f b path/to/repo_b.git
git remote update
git diff master remotes/b/master
git remote rm b
```

Source: <https://stackoverflow.com/questions/1968512/getting-the-difference-between-two-repositories>

## How do I "commit" changes in a git submodule?

So, first commit/push your submodule's changes:

```
cd path/to/submodule
git add <stuff>
git commit -m "comment"
git push
```

Then, update your main project to track the updated version of the submodule:

```
cd /main/project
git add path/to/submodule
git commit -m "updated my submodule"
git push
```

Source: <https://stackoverflow.com/questions/5542910/how-do-i-commit-changes-in-a-git-submodule>

## Remove sensitive files and their commits from Git history

```
git filter-branch --force --index-filter "git rm -r --cached --ignore-unmatch PATH-TO-YOUR-FILE-WITH-SENSITIVE-DATA" --prune-empty --tag-name-filter cat -- --all
git push --force --verbose --dry-run
git push --force
```

Source:

<https://stackoverflow.com/questions/872565/remove-sensitive-files-and-their-commits-from-git-history>

## How to merge two or multiple git repositories into one

- <https://medium.com/altcampus/how-to-merge-two-or-multiple-git-repositories-into-one-9f8a5209913f>

- <https://stackoverflow.com/questions/1425892/how-do-you-merge-two-git-repositories>
- <https://github.com/apenwarr/git-subtree>

From:

<https://wiki.xw3.org/> - **wiki.xw3.org**

Permanent link:

<https://wiki.xw3.org/git>

Last update: **2025-03-18**

