

Resource

Online

Official Website

Download

GitCheatSheet [.pdf]
Git cheat sheet, extended edition prepa
red [.svg]
Git Quick Reference [.pdf]
Git Cheat Sheet In Chinese

Related

Bazaar
CVS
SVN

Remote Update / Publish

List remotes

git remote -v

Show information

git remote show remote

Add remote

git remote add path/url

Fetch changes

git fetch remote

Fetch + merge

git pull remote branch

Publish local to remote

git push remote branch

Delete remote branch

git push remote :branch

Publish tags

git push origin/upstream --tags

Create Git

From existing directory

```
cd project_dir
git init
git add .
```

From other repository

```
git clone existing_dir new_dir
git clone git://github.com/user/repo.g
it
git clone https://github.com/user/rep
o.git
```

Merge/Rebase

Merge branch into current

git merge branch

Rebase into branch

```
git rebase branch
git rebase master branch
```

Abort rebase

git rebase --abort

Merge tool to solve conflicts

git mergetool

To view the merge conflicts

```
git diff
  complete conflict diff
git diff --base $file
  against base file
git diff --ours $file
  against your changes
git diff --theirs $file
  against other changes
```

To discard conflicting patch

```
git reset --hard
git rebase --skip
```

After resolving conflicts

```
git add $conflicting_file
  do for all resolved files
git rebase --continue
```

Local Changes

Changed in working directory

git status

Tracked file changes

git diff

Add changed files

git add file1 file2 file3

Remove file

```
git rm file
git rm dir/ -r
  (recursive under directory)
```

See files ready for commit

git diff --cached

Commit changes

```
git commit
git commit -m "My message"
git commit -a -m "My Message"
  (tracked files only, auto add)
```

Change last commit

git commit --amend

Revert changes to file

git checkout -- file

Revert changes (new commit)

git revert HEAD

Return to last committed state

git reset --hard HEAD

Useful Commands

Finding Regressions

git bisect start

```
  to start
git bisect good $id
```

```
  $id is the last working version
git bisect bad $id
```

```
  $id is a broken version
git bisect bad/good
```

```
  to mark it as bad or good
git bisect visualize
```

```
  to launch gitk and mark it
git bisect reset
```

```
  once you're done
```

Check for Errors and Cleanup R epository

```
git fsck
git gc --prune
```

Search Working Directory for foo

```
()
git grep "foo()"
```

History

Show all commits

git log

Short Format

git log --pretty=--short

Patches

git log -p

Show file commits

git log file

Show directory commits

git log dir/

Stats

git log --stat

Who changed file

git blame file

Branching/Tagging

List branches

git branch

Switch to branch

git checkout branch

Create new branch

git branch new

Create branch from existing

git branch new existing

Delete branch

git branch -d branch

Tag current commit

git tag tagname

Ad