

GET GIT

FORK OTHER VERSION CONTROL

This cheat sheet's like Game Genie for Git; it'll help you reach the next level without breaking a sweat. If you need other info, you can find more docs at [Git's website](#).

MAKE CHANGES

Review edits and craft a commit transaction.

<code>\$ git status</code>	Lists all new or modified files to be committed
<code>\$ git diff</code>	Shows file differences not yet staged
<code>\$ git add <i>file</i></code>	Snapshots the file in preparation for versioning
<code>\$ git diff --staged</code>	Shows file differences between staging and the last file version
<code>\$ git reset <i>file</i></code>	Unstages the file, but preserves its contents
<code>\$ git commit -m "<i>descriptive message</i>"</code>	Records file snapshots permanently in version history

CONFIGURE TOOLING

Configure user information for all local repositories

<code>\$ git config --global user.name "<i>name</i>"</code>	Sets the name you want attached to your commit transactions
<code>\$ git config --global user.email "<i>email address</i>"</code>	Sets the email you want attached to your commit transactions
<code>\$ git config --global color.ui auto</code>	Enables helpful colorization of command line output

CREATE REPOSITORIES

Start a new repository or obtain one from an existing URL

<code>\$ git init <i>project-name</i></code>	Creates a new local repository with the specified name
--	--

```
$ git clone url
```

Downloads a project and its entire version history

GROUP CHANGES

Name a series of commits and combine completed efforts

```
$ git branch
```

Lists all local branches in the current repository

```
$ git branch branch-name
```

Switches to the specified branch and updates the working directory

```
$ git merge branch
```

Combines the specified branch's history into the current branch

```
$ git branch -d branch-name
```

Deletes the specified branch

SYNCHRONIZE CHANGES

Register a repository bookmark and exchange version history

```
$ git fetch bookmark
```

Downloads all history from the repository bookmark

```
$ git merge bookmark/branch
```

Combines bookmark's branch into current local branch

```
$ git push alias branch
```

Uploads all local branch commits to GitHub

REFACTOR FILENAMES

Relocate and remove versioned files

```
$ git rm file
```

Deletes the file from the working directory and stages the deletion

```
$ git rm --cached file
```

Removes the file from version control but preserves the file locally

```
$ git mv file-original file-renamed
```

Changes the file name and prepares it for commit

SAVE FRAGMENTS

Shelve and restore incomplete changes

<code>\$ git stash</code>	Temporarily stores all modified tracked files
<code>\$ git stash list</code>	Lists all stashed changesets
<code>\$ git stash pop</code>	Restores the most recently stashed files
<code>\$ git stash drop</code>	Discards the most recently stashed changeset

REDO COMMITS

Erase mistakes and craft replacement history

<code>\$ git reset <i>commit</i></code>	Undoes all commits after <i>commit</i> , preserving changes locally
<code>\$ git reset --hard <i>commit</i></code>	Discards all history and changes back to the specified commit

REVIEW HISTORY

Browse and inspect the evolution of project files

<code>\$ git log</code>	Lists version history for the current branch
<code>\$ git log --follow <i>file</i></code>	Lists version history for a file, including renames
<code>\$ git diff <i>first-branch...second-branch</i></code>	Shows content differences between two branches
<code>\$ git show <i>commit</i></code>	Outputs metadata and content changes of the specified commit