

GIT ESSENTIALS - Complete Developer's Cheatsheet

1. INSTALLATION & SETUP

Platform	Command
macOS	<code>brew install git</code>
Ubuntu/Debian	<code>sudo apt-get install git</code>
Windows	Download from git-scm.com

Initial Configuration:

```
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
git config --list # View settings
```

2. ESSENTIAL DAILY COMMANDS

Command	Description
<code>git status</code>	Check current status
<code>git add .</code>	Stage all changes
<code>git commit -m "msg"</code>	Commit with message
<code>git push</code>	Upload to remote
<code>git pull</code>	Download from remote
<code>git log --oneline</code>	View history
<code>git diff</code>	Show unstaged changes

3. REPOSITORY OPERATIONS

Task	Command
Initialize repo	<code>git init</code>
Clone repo	<code>git clone <url></code>
Add remote	<code>git remote add origin <url></code>
View remotes	<code>git remote -v</code>

4. BRANCHING & MERGING

Operation	Command
List branches	<code>git branch</code>
Create branch	<code>git branch feature-name</code>
Create & switch	<code>git checkout -b feature-name</code>
Switch branch	<code>git checkout branch-name</code>
Merge branch	<code>git merge feature-name</code>
Delete branch	<code>git branch -d feature-name</code>

5. UNDOING CHANGES

Scenario	Command
Unstage file	<code>git restore --staged file.py</code>
Discard changes	<code>git restore file.py</code>
Amend last commit	<code>git commit --amend -m "New"</code>
Undo commit (keep)	<code>git reset HEAD~1</code>
Undo commit (discard)	<code>git reset --hard HEAD~1</code>

6. VIEWING HISTORY

Task	Command
Commit history	<code>git log --oneline</code>
Graph view	<code>git log --graph --all</code>
Show commit	<code>git show <sha></code>
File history	<code>git log --follow file.py</code>
Who changed	<code>git blame file.py</code>

7. GIT STASH

Operation	Command
Save changes	<code>git stash</code>
List stashes	<code>git stash list</code>
Apply stash	<code>git stash apply</code>
Apply & remove	<code>git stash pop</code>
Clear all	<code>git stash clear</code>

8. BEST PRACTICES

DO	DONT
----	------

Commit often	Make huge commits
Meaningful messages	Vague messages
Pull before push	Push without checking
Use branches	Work on main directly
Review changes	Blindly commit all
Use .gitignore	Commit secrets

Pro Tip: Master git status, git add, git commit, git push - You're 80% there!

Remember: Git is your safety net. Don't be afraid to experiment!

Created by: enriktech.com | Contact: @enriktech.com | Version: February 2026