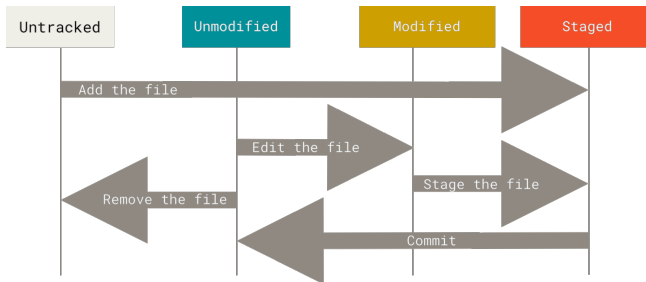


# Git Cheat Sheet

## Workflow



## Start a new repository

Command	Description
<code>git init</code>	Create a new repository in the current directory
<code>git clone &lt;url&gt; [&lt;dir&gt;]</code>	Clone a remote repository

## History

Command	Description
<code>git log</code>	Show commit logs
<code>git log --oneline</code>	Show commit logs with one line per commit
<code>git show</code>	Show the changes of the last commit
<code>git shortlog</code>	Summarize git log output by author
<code>git blame &lt;file&gt;</code>	Show who changed which line in a file

## Local changes

Command	Description
<code>git status</code>	List which files are staged, unstaged, and untracked
<code>git add &lt;file&gt;</code>	Add a file to the staging area
<code>git add -p</code>	Interactively choose hunks of a file to add to the staging area

## Local changes (cont.)

Command	Description
<code>git diff</code>	Show unstaged changes
<code>git diff --staged</code>	Show staged changes
<code>git commit</code>	Commit staged changes using the default editor
<code>git commit -m &lt;message&gt;</code>	Commit staged changes with a message provided on the command line
<code>git commit --amend</code>	Amend the last commit
<code>git reset HEAD &lt;file&gt;</code>	Unstage a file
<code>git checkout -- &lt;file&gt;</code>	Discard local changes in a file

## Branches

Command	Description
<code>git branch</code>	List local branches
<code>git branch -r</code>	List remote branches
<code>git branch &lt;branch&gt;</code>	Create a new branch
<code>git checkout &lt;branch&gt;</code>	Switch to a branch
<code>git checkout -b &lt;branch&gt;</code>	Create a new branch and switch to it
<code>git merge &lt;branch&gt;</code>	Merge a branch into the active branch
<code>git branch -d &lt;branch&gt;</code>	Delete a branch

## Tags

Command	Description
<code>git tag</code>	List tags
<code>git tag &lt;tag&gt;</code>	Create a tag
<code>git tag -a &lt;tag&gt;</code>	Create an annotated tag
<code>git tag -s &lt;tag&gt;</code>	Create a GPG-signed tag
<code>git tag -d &lt;tag&gt;</code>	Delete a tag
<code>git show &lt;tag&gt;</code>	Show the tag data

## Collaboration

Command	Description
<code>git remote -v</code>	List remote repositories
<code>git remote add &lt;name&gt; &lt;url&gt;</code>	Add a new remote repository
<code>git fetch &lt;remote&gt;</code>	Download all changes from a remote repository but don't integrate them
<code>git pull &lt;remote&gt;</code>	Download changes and directly merge/integrate them
<code>git pull &lt;remote&gt; &lt;branch&gt;</code>	Download changes and directly merge/integrate a remote branch
<code>git push &lt;remote&gt; &lt;branch&gt;</code>	Upload local changes to a remote repository
<code>git push &lt;remote&gt; --delete &lt;branch&gt;</code>	Delete a remote branch
<code>git push &lt;remote&gt; --tags</code>	Push tags to a remote repository

## Stashing

Command	Description
<code>git stash</code>	Stash the changes in a dirty working directory away
<code>git stash list</code>	List all stashes
<code>git stash pop</code>	Apply the last stash and delete it
<code>git stash apply</code>	Apply the last stash but don't delete it
<code>git stash drop</code>	Delete the last stash
<code>git stash clear</code>	Delete all stashes

## Writing good commit messages

1. Separate subject from body with a blank line
2. Limit the subject line to 50 characters
3. Capitalize the subject line
4. Do not end the subject line with a period
5. Use the imperative mood in the subject line
6. Wrap the body at 72 characters
7. Use the body to explain what and why vs. how