

Git Cheat Sheet



Git Basics

<code>git init</code> <code>git commit -m</code>	Initialize a repository or commit changes that are ready to be shared. Use <code>git commit</code> to save all changes that you've made.
<code>git clone <url></code>	Clone a repository or create a local repository. Clone a repository to work on. Clone a repository to create a local repository. <code>git clone <url></code>
<code>git diff</code> <code>git diff --staged</code> <code>git diff --cached</code>	Show the changes that are staged for commit. Show the changes that are staged for commit. Show the changes that are staged for commit.
<code>git add</code> <code>git commit</code>	Stage changes to be committed. Commit the changes to the repository.
<code>git status</code> <code>git log</code>	Check the status of the repository. Show the commit history.
<code>git checkout</code>	Switch to a different branch or commit.
<code>git merge</code>	Merge the changes from one branch into another.
<code>git pull</code>	Fetch the changes from the remote repository and merge them into the current branch.
<code>git push</code>	Push the changes from the local repository to the remote repository.

Working Changes

<code>git add</code> <code>git commit</code>	Commit the changes to the repository. Use <code>git add</code> to stage changes and <code>git commit</code> to commit them.
<code>git diff</code> <code>git diff --staged</code>	Show the changes that are staged for commit. Show the changes that are staged for commit.
<code>git status</code>	Check the status of the repository.

Reverting Git History

<code>git reset --hard</code>	Reset the repository to the specified commit. Use <code>git reset --hard</code> to reset the repository to the specified commit.
<code>git revert</code>	Revert the changes from a specific commit. Use <code>git revert</code> to revert the changes from a specific commit.
<code>git checkout</code>	Switch to a different branch or commit.

Git Branches

<code>git branch</code>	Create a new branch. Use <code>git branch</code> to create a new branch.
<code>git checkout</code> <code>git merge</code>	Switch to a different branch or merge the changes from one branch into another.
<code>git push</code> <code>git pull</code>	Push the changes from the local repository to the remote repository or pull the changes from the remote repository.

Remote Repositories

<code>git remote add</code> <code>git remote rm</code>	Add a new remote repository or remove a remote repository. Use <code>git remote add</code> to add a new remote repository and <code>git remote rm</code> to remove a remote repository.
<code>git fetch</code> <code>git pull</code>	Fetch the changes from the remote repository or pull the changes from the remote repository.
<code>git push</code> <code>git push --force</code>	Push the changes from the local repository to the remote repository or push the changes from the local repository to the remote repository with force.

Additional Options +



git checkout

<code>git checkout -b <branch></code>	Define the branch name from scratch and switch to the new branch
<code>git checkout -b <branch> <commit></code>	Define the branch name from scratch and switch to the new branch
<code>git checkout -b <branch> <branch></code>	Create a new branch from an existing branch and switch to it
<code>git checkout -b <branch> <commit></code>	Create a new branch from an existing branch and switch to it
<code>git checkout -b <branch> <branch></code>	Create a new branch from an existing branch and switch to it
<code>git checkout -b <branch> <commit></code>	Create a new branch from an existing branch and switch to it

git log

<code>git log --oneline</code>	Show the commit history in a compact way
<code>git log --graph</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline --abbrev=7</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline --abbrev=7 --max-count=10</code>	Show the commit history with a graphical representation

<code>git log --graph --oneline --abbrev=7 --max-count=10</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline --abbrev=7 --max-count=10 --format=%h</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline --abbrev=7 --max-count=10 --format=%h</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline --abbrev=7 --max-count=10 --format=%h</code>	Show the commit history with a graphical representation
<code>git log --graph --oneline --abbrev=7 --max-count=10 --format=%h</code>	Show the commit history with a graphical representation

git diff

<code>git diff</code>	Show differences between working directory and last commit
<code>git diff --cached</code>	Show differences between staged changes and last commit

git merge

<code>git merge</code>	Integrate changes from another branch into the current branch
<code>git merge --no-commit</code>	Integrate changes from another branch into the current branch
<code>git merge --no-commit --no-ff</code>	Integrate changes from another branch into the current branch
<code>git merge --no-commit --no-ff --no-edit</code>	Integrate changes from another branch into the current branch
<code>git merge --no-commit --no-ff --no-edit --no-ff</code>	Integrate changes from another branch into the current branch

git rebase

<code>git rebase</code>	Rebase the current branch onto the specified branch
-------------------------	---

git push

<code>git push</code>	Push the current branch to the remote repository
-----------------------	--

git pull

<code>git pull</code>	Fetch the remote repository and merge the changes into the current branch
<code>git pull --rebase</code>	Fetch the remote repository and rebase the changes onto the current branch
<code>git pull --rebase --no-commit</code>	Fetch the remote repository and rebase the changes onto the current branch
<code>git pull --rebase --no-commit --no-ff</code>	Fetch the remote repository and rebase the changes onto the current branch