Git Setup

git init [directory]	create a Git repository from an existing directory
git clone [repo / URL]	clone / download a repository onto local machine
git clone [URL] [folder]	clone a repository from a remote location into a specified folder [folder] on your local machine

Git Branches

git branch	list all branches in the repository
git branch -a	list all remote branches
git branch [branch]	create a new branch under the specified name
git checkout [branch]	switch to another branch (either an existing one or by creating a new one under the specified name]
git branch -d [branch]	delete a local branch
git branch -m [new_branch_name]	rename the branch you are currently working in
git merge [branch]	merge the specified branch with the current branch

Undoing Changes

git revert [file/directory]	undo all changes in the specified file/directory by creating a new commit and applying it to the current branch
git reset [file]	unstage the specified file without overwriting changes
git reset [commit]	undo all changes that happened after the specified commit
git clean -n	see which files should be removed from the current directory
git clean -f	remove the unnecessary files in the directory

Git Configuring

git config --global user. name "[your_name]"	set an author name that will be attached to all commits by the current user
git config --global user. email " [email_address]"	set an email address that will be attached to all commits by the current user
git config --global color.ui auto	set Git's automatic command line coloring
git config --global alias. [alias_name] [git_command]	create a shortcut (alias) for a Git command
git config --system core.editor [text_editor]	set a default text editor for all the users on the machine
git config --global --edit	open Git's global configuration file

Rewriting History

git commit --amend	replace the last commit with a combination of the staged changes and the last commit combined
git rebase [base]	rebase the current branch with the specified base (it can be a branch name, tag, reference to a HEAD, or a commit ID)
git reflog	list changes made to the HEAD of the local repository

Making Changes

git add [file/directory]	stage changes for the next commit
git add .	stage everything in the directory for an initial commit
git commit -m " [descriptive_message]"	commit the previously staged snapshot in the version history with a descriptive message included in the command

Managing Files

git status	show the state of the current directory (along with staged, unstaged, and untracked files)
git log	list the complete commit history of the current branch
git log --all	list all commits from all branches
git log [branch1]..[branch2]	show which commits are on the first branch, but not on the second one
git diff	see the difference between the working directory and the index (which changes have not been commited yet)
get diff --cached	see the difference between the last commit and the index
get diff HEAD	see the difference between the last commit and the working directory
git show [object]	show the content and metadata of an object (blob, tree, tag, or commit)

Remote Repositories

git remote add [name] [URL]	create a new connection to a remote repository and give it a name to serve as a shortcut to the URL
git fetch [remote_repo] [branch]	fetch a branch from a remote repository
git pull [remote_repo]	fetch the specified repository and merge it with the local copy
git push [remote_repo] [branch]	push a branch to a remote repository with all its commits and objects