



Git Cheat Sheet

@nmelgar

About Git

What is Git?:

Git is a distributed version control system that allows you to track changes to files over time. It's designed to be fast, scalable, and ideal for both small and large projects.

Local Repository

With Git, you start by creating a local repository on your machine. This repository contains all the files and the complete history of changes made to those files.

Commits

As you make changes to your files, Git allows you to create commits. A commit is a snapshot of the changes you've made at a particular point in time. Each commit has a unique identifier called a commit hash.

Branches

Branches, are separate lines of development. They enable you to work on different features or bug fixes without affecting the main codebase. You can switch between branches and merge changes from one branch to another.

Staging Area

Git has a staging area, also known as the index, where you can choose which changes to include in your next commit. This allows you to have more control over what gets committed.

Remote Repository

A remote repository is a Git repository hosted on a remote server. GitHub is a popular platform for hosting remote repositories. It provides additional collaboration features and a web-based interface for managing Git repositories.

GitHub

GitHub is a web-based platform that uses Git for version control. It offers a graphical interface to interact with Git repositories, making it easier to collaborate with others. GitHub allows you to host your remote repositories, share code with others, and contribute to open-source projects.

Pushing and Pulling

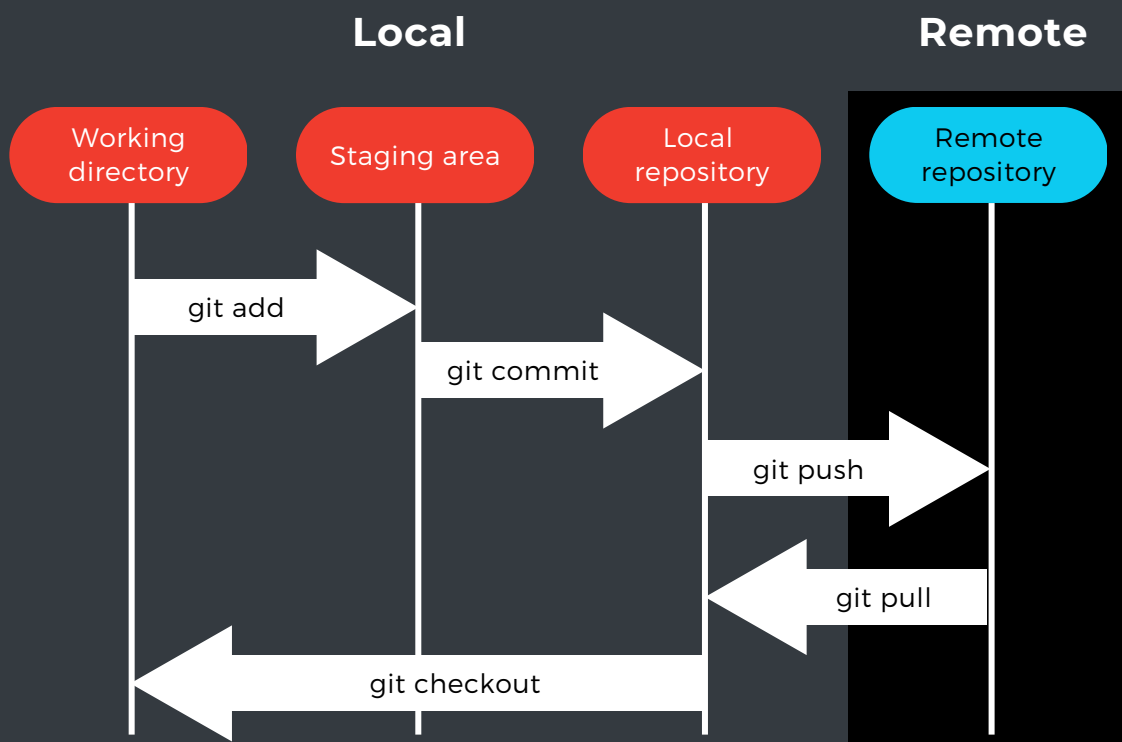
To synchronize your local repository with a remote repository on GitHub, you use the `git push` command to upload your local commits. Conversely, you use the `git pull` command to fetch and merge the latest changes from the remote repository to your local repository.

Git workflow (simplified)

1. Initialize a Git repository on your local machine using **git init**.
2. Make changes to your files and use **git add** to stage the changes.
3. Create a commit using **git commit** to save the staged changes with a commit message.
4. Push your commits to a remote repository on GitHub using **git push**.
5. Collaborate with others by cloning their remote repositories using **git clone** or by creating branches, making changes, and merging them back using **git pull** and **git merge**.

Git and GitHub provide a powerful and flexible version control system. This is just a high-level overview, there are many additional concepts and commands in Git that can be explored in more detail.

Git Illustrated



Git commands

Git has a collection of commands that you can use to perform different tasks. Each command serves a specific function, like creating a new repository, tracking changes, merging code, and more.

Git commands has subcommands and options:

Subcommand	<p>An additional keyword that follows the main Git command. It specifies a specific action or operation you want to perform with that command. They allow you to extend the functionality of the main command by defining different tasks/actions related to it.</p> <p>e.g. in the command git remote add, remote is the main command, and add is the subcommand.</p>
Option	<p>AKA a flag or switch, is used to modify the behavior of a Git command. Usually represented by a hyphen (-) followed by a single character or a double hyphen (--) followed by a descriptive name. They are used to modify how the command operates without changing its core functionality.</p> <p>e.g. in the command git remote -v, -v is the option. It specifies the "verbose" output and instructs Git to provide more detailed information along with the default output.</p>

Common Git commands

git init	<p>Initializes a new Git repository in the current directory.</p> <p>git init -> creates a new empty Git repository.</p>
git clone	<p>Copies a remote repository and sets it up locally.</p> <p>git clone <repository_url> -> clones the remote repository specified by <repository_url> to your local machine.</p>
git add	<p>Adds changes or new files to the staging area.</p> <p>git add <file_name> -> stages the changes made to <file_name> for the next commit.</p>
git commit	<p>Records the changes to the repository.</p> <p>git commit -m "Commit message" -> commits the staged changes with the provided commit message.</p>
git push	<p>Uploads the local commits to a remote repository.</p> <p>git push origin master -> pushes the commits from the local "master" branch to the remote repository named "origin."</p>

git pull	Fetches changes from a remote repository and integrates them into the current branch. git pull origin master -> fetches changes from the remote repository named "origin" and merges them into the local "master" branch.
git branch	Lists, creates, or deletes branches. git branch -> lists all the branches in the repository.
git checkout	Switches between branches or restores files from a previous commit. git checkout <branch_name> -> switches to the branch specified by <branch_name>.
git merge	Combines changes from different branches. git merge <branch_name> -> merges the changes from <branch_name> into the current branch.
git status	Shows the current state of the repository. git status -> displays information about the modified, staged, and untracked files in the repository.
git log	Displays the commit history of the repository. git log -> shows a chronological list of commits, including commit hashes, authors, dates, and commit messages.
git diff	Shows the differences between the current state and previous commits or between branches. git diff -> displays the changes made to the files in the current working directory compared to the last commit.
git stash	Temporarily saves changes that are not ready to be committed. git stash save "feature in progress" -> stores the current changes in a stash with the provided message.
git remote	Manages remote repositories. git remote -v lists the configured remote repositories and their URLs. git remote add <remote_name> <repository_url> -> adds a new remote repository
git fetch	Downloads the latest changes from a remote repository without merging them. git fetch origin -> fetches the latest changes from the remote repository named "origin" without merging them into the local branch.
git reset	Undoes commits by moving the branch pointer backward. git reset HEAD~1 -> undoes the last commit and moves the branch pointer one commit back. (HEAD~1 refers to the commit one step before)
git rebase	Applies changes from one branch onto another branch. git rebase <branch_name> -> incorporates the changes from <branch_name> onto the current branch.
git blame	Shows who last modified each line of a file. git blame <file_name> -> displays the author and commit information for each line of <file_name>.
git config	Sets up Git configuration variables. git config --global user.name "John Doe" -> sets the global username to "John Doe" for all Git repositories on your machine.