|  |  |
| --- | --- |
| **Git Command** | **Meaning** |
| git init | Initializes a new Git repository in the current directory. This creates a hidden .git folder that stores all the repository's metadata. |
| git clone <repository\_url> | Creates a local copy of a remote Git repository. This downloads all the project's files and its complete history. |
| git add <file\_name> | Stages a specific file for the next commit. This tells Git that you want to include changes to this file in the next snapshot. |
| git add . | Stages all changes in the current directory for the next commit. This is a convenient way to stage multiple modified or new files. |
| git commit -m "Your commit message" | Records the staged changes as a new commit in the repository's history. The commit message should be a concise and descriptive summary of the changes. |
| git status | Shows the current state of the working directory and the staging area. It tells you which files are modified, staged, or untracked. |
| git diff | Shows the differences between the working directory and the staging area (changes that are not yet staged). |
| git diff --staged | Shows the differences between the staging area and the last commit (changes that are staged but not yet committed). |
| git log | Displays a history of commits for the current branch. Each entry shows the commit hash, author, date, and commit message. |
| git log --oneline | Displays a condensed, single-line version of the commit history. |
| git branch | Lists all local branches in the repository. The currently active branch is usually highlighted. |
| git branch <new\_branch\_name> | Creates a new branch with the specified name, pointing to the current commit. |
| git checkout <branch\_name> | Switches to the specified branch. This changes your working directory to reflect the files as they were in that branch. |
| git checkout -b <new\_branch\_name> | Creates a new branch and immediately switches to it. This is a shortcut for git branch <new\_branch\_name> followed by git checkout <new\_branch\_name>. |
| git merge <branch\_to\_merge> | Integrates changes from one branch into the current branch. This combines the commit histories of the two branches. |
| git pull | Fetches changes from a remote repository and integrates them into the current local branch. It's a combination of git fetch and git merge. |
| git push <remote\_name> <branch\_name> | Uploads your local commits to a remote repository. Typically, remote\_name is origin and branch\_name is main or master. |
| git remote -v | Lists the remote repositories that your local repository is connected to. |
| git remote add <remote\_name> <remote\_url> | Adds a new remote repository to your local configuration. |
| git reset --hard <commit\_hash> | Resets the current branch to a specific commit, discarding all changes in the working directory and staging area after that commit. Use with extreme caution as it's destructive. |
| git revert <commit\_hash> | Creates a new commit that undoes the changes introduced by a specific commit. This is a safer way to undo changes as it preserves the history. |
| git stash | Temporarily saves changes that are not ready to be committed. This allows you to switch branches or perform other operations without committing incomplete work. |
| git stash pop | Applies the most recently stashed changes and removes them from the stash list. |
| git fetch | Downloads objects and refs from another repository. It retrieves all the new information from the remote repository but doesn't merge it into your local branches. |
| git tag <tag\_name> | Creates a lightweight tag at the current commit. Tags are used to mark specific points in history, typically for releases. |
| git rebase <base\_branch> | Rewrites a series of commits on top of another base tip. This can lead to a cleaner, linear history but should be used carefully, especially on shared branches. |
| git rm <file\_name> | Removes a file from the working directory and stages the deletion. |
| git mv <old\_file\_name> <new\_file\_name> | Renames or moves a file and stages the change. |

Export to Sheets

**Understanding Key Git Concepts:**

* **Repository (Repo):** A .git directory containing all the project's files, commit history, and metadata.
* **Working Directory:** The actual files you see and edit on your computer.
* **Staging Area (Index):** An intermediate area where you prepare changes before committing them.
* **Commit:** A snapshot of your repository at a specific point in time. Each commit has a unique ID (hash) and a commit message.
* **Branch:** A lightweight movable pointer to a commit. Branches allow you to work on different features or bug fixes concurrently without affecting the main codebase.
* **Master/Main Branch:** The default and primary branch in most Git repositories, typically representing the stable version of the code.
* **Remote:** A version of your repository hosted on the internet (e.g., GitHub, GitLab, Bitbucket). origin is the default name for the remote repository your local repo was cloned from.