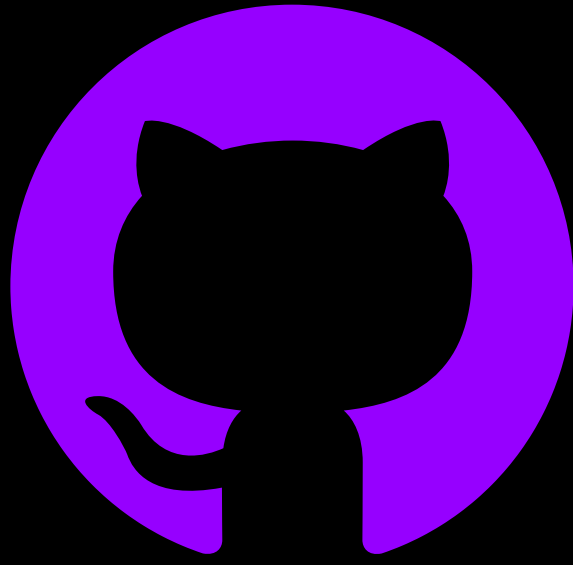




Kishore M
@kishorem0077

01



Master GitHub with Ultimate Cheatsheet!





Initializing a New Repository

to create a new Git repository.

```
git init
```

Adding an Existing Repository to a Project

to add a remote repository to your existing project

```
git remote add origin https://github.com/username/repo.git
```

Cloning a Repository

to clone a repository from a remote URL

```
git clone https://github.com/username/repo.git
```





Check Repository Status

Check the status of your repository, showing staged, unstaged, and untracked files

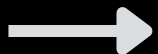
```
git status
```

View the commit history

```
git log
```

Show changes between commits, commit and working tree, etc

```
git diff
```





Create and Delete Branches

List all branches

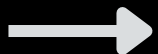
```
git branch
```

Create a new branch

```
git branch feature-branch
```

Delete a branch

```
git branch -d feature-branch
```





Switch Branches

Create and switch to a new branch

```
git checkout -b feature-branch
```

Switch to an existing branch

```
git checkout main
```

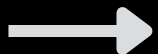
Stage Changes

Add a file to the staging area

```
git add filename.txt
```

Add all changes to the staging area

```
git add .
```





Commit Changes

Commit changes with a message

```
git commit -m "Initial commit"
```

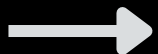
Stage and commit all changes

```
git commit -a -m "Updated all files"
```

Amend a Commit

Amend the previous commit with a new message

```
git commit --amend -m "Updated commit message"
```





View Commit History

Show commit logs

```
git log
```

Show commit logs in one line each

```
git log --oneline
```

Show commit logs as a graph

```
git log --graph
```





Revert Commits

Create a new commit that undoes changes from a previous commit

```
git revert [commit-hash]
```

Move the current branch tip backward to the specified commit

```
git reset [commit-hash]
```

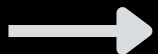
Undo Changes

Discard changes in the working directory

```
git checkout -- filename.txt
```

Unstage a file without changing its content

```
git reset HEAD filename.txt
```





Push Changes

Push changes to the remote repository

```
git push
```

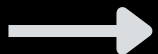
Push a specific branch to the remote repository

```
git push origin feature-branch
```

Force Push Changes

Force push changes (use with caution)

```
git push --force
```





Pull Changes

Fetch and merge changes from the remote repository

```
git pull
```

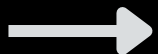
Pull changes from a specific branch

```
git pull origin feature-branch
```

Revert Pull

Revert to the state before the last pull

```
git reset --hard ORIG_HEAD
```





Stash Changes

Stash your changes for later

```
git stash
```

List all stashes

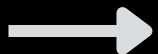
```
git stash list
```

Apply the latest stash

```
git stash apply
```

Remove the latest stash

```
git stash drop
```





Create and Apply Tags

Create a new tag

```
git tag v1.0
```

List all tags

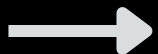
```
git tag
```

Delete a tag

```
git tag -d v1.0
```

Push a tag to the remote repository

```
git push origin v1.0
```





Merge Branches

Merge a branch into the current branch

```
git merge feature-branch
```

Resolve Merge Conflicts

- **Conflict Markers:** Look for `<<<<<<<`, `====`, and `>>>>>>>` to find conflicts.
- **Resolve Conflicts:** Edit the conflicting files to resolve conflicts.

Add the resolved files

```
git add filename.txt
```

Commit the changes

```
git commit
```





Remote Repositories

Add a remote repository

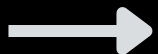
```
git remote add origin https://github.com/username/repo.git
```

List all remote repositories

```
git remote -v
```

Remove a remote repository

```
git remote remove origin
```





Kishore M
@kishorem0077

15

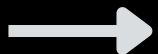
Fetch Changes

Download objects and refs from another repository

```
git fetch
```

Fetch from the remote repository named "origin"

```
git fetch origin
```





Rebase Branches

Reapply commits on top of another base tip

```
git rebase main
```

Abort the rebase process

```
git rebase --abort
```

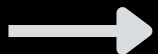
Continue the rebase process after resolving conflicts

```
git rebase --continue
```

Cherry-pick Commits

Apply the changes from a specific commit onto the current branch

```
git cherry-pick [commit-hash]
```





Ignoring Files

- **.gitignore:** Create a .gitignore file to specify untracked files to ignore
- **Example**

```
# Ignore all .log files  
*.log
```

Removing a Repository from a Project

Remove the Git repository from your project

```
rm -rf .git
```

-- or --

```
rmdir /s .git
```

This deletes the .git directory and all its contents, effectively "de-git" your project

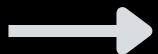
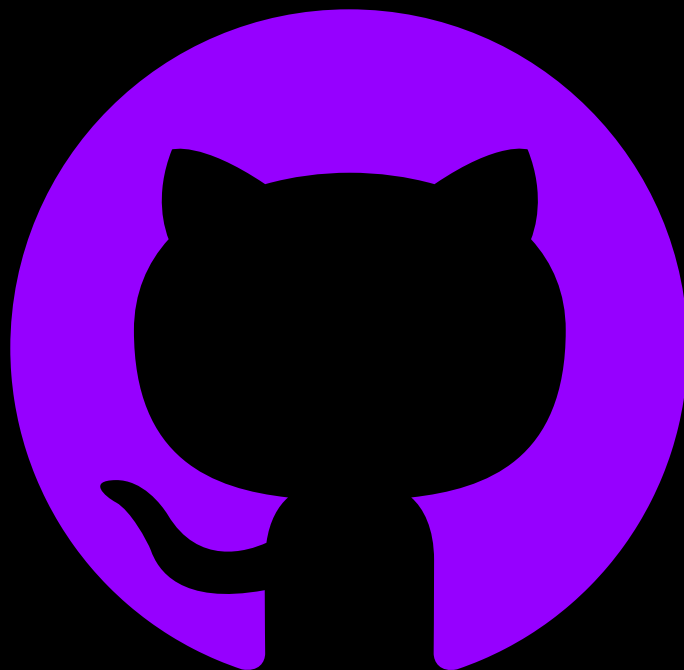




Summary & Resources

You now have the essential commands to navigate GitHub like a pro! For more detailed guides, check out the following resources

- **Git Documentation** - <https://git-scm.com/doc>
- **GitHub Guides** - <https://docs.github.com/en>



Did you Find it Useful?



@kishorem0077
Kishore M

Follow for more !

