



Education  
@topdev\_media



# GitHub

# CHEAT SHEET



## Configuration and Setup

Sets the username to be used for all commits on the machine.

```
git config --global user.name "[name]"
```

Sets the email address to be used for all commits on the machine.

```
git config --global user.email "[email address]"
```

## Starting a Project

Initializes a new local repository.

```
git init [project-name]
```

Clones an existing project and its version history.

```
git clone [url]
```

## Changes and Staging

Shows the status of modified files in the working directory.

```
git status
```

Adds a file to the staging area.

```
git add [file]
```

Adds all modified files to the staging area.

```
git add .
```



Commits the staged files with a descriptive message.

```
git commit -m "[commit message]"
```

## Managing Commits

Modifies the last commit.

```
git commit --amend
```

Modifies the last commit without changing the commit message.

```
git commit --amend --no-edit
```

Displays a condensed commit history.

```
git log --oneline
```

Shows the commit history in graphical form.

```
git log --graph
```

Reapplies changes on top of another base tip.

```
git rebase [branch]
```

Creates a new commit that undoes the specified changes.

```
git revert [commit]
```



## Branches and Merging

Lists all branches in the repository.

```
git branch
```

Creates a new branch.

```
git branch [branch-name]
```

Switches to the specified branch.

```
git checkout [branch-name]
```

Merges the specified branch's history into the current branch.

```
git merge [branch]
```

Deletes the specified branch.

```
git branch -d [branch-name]
```

## Sharing and Updating Projects

Uploads local branch commits to the remote repository branch.

```
git push [alias] [branch]
```

Integrates remote branch changes into the current local branch.

```
git pull
```

## Inspection & Comparison

Displays the version history for the current branch.

```
git log
```

Shows changes to a specific file.

```
git log --follow [file]
```

Shows content differences between two branches.

```
git diff [branchB]
```

## Working with Specific Commits

Shows metadata and content changes of the specified commit.

```
git show [commit]
```

Applies the changes introduced by the specified commit on the current branch.

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

Moves the current branch tip backward to the specified commit, leaving the changes in staging.

```
git reset --soft [commit]
```

Moves the current branch tip backward to the specified commit, leaving the changes unstaged.

```
git reset [commit]
```



Resets the working directory to the specified commit, discarding all changes.

```
git reset --hard [commit]
```

## Undoing Changes

Undoes all commits after [commit], preserving changes locally.

```
git reset [commit]
```

Erases all changes after the specified commit.

```
git reset --hard [commit]
```



**Education**  
@topdev\_media



This cheatsheet provides an overview of the essential Git commands for effectively managing your projects.

Use it as a **quick reference** for your daily versioning and collaboration tasks with Git.

**Save it for later !**





**Follow for more**



Halim Shams