

# Git Cheat Sheet

## Basic Commands

### 1. git init

- Initialise a new Git repository.
- Example: `git init`

### 2. git clone [url]

- Clone a repository into a new directory.
- Example: `git clone https://github.com/user/repo.git`

### 3. git add [file]

- Add a file to the staging area.
- Example: `git add example.txt`

### 4. git commit -m "[commit message]"

- Commit changes with a message.
- Example: `git commit -m "Initial commit"`

### 5. git status

- Show the working tree status.
- Example: `git status`

### 6. git pull [remote] [branch]

- Fetch from and integrate with another repository or a local branch.
- Example: `git pull origin main`

### 7. git push [remote] [branch]

- Update remote refs along with associated objects.
- Example: `git push origin main`

# Branching and Merging

## 1. git branch

- List, create, or delete branches.
- Example: `git branch new-feature`

## 2. git branch -d [branch name]

- Delete a branch.
- Example: `git branch -d old-feature`

## 3. git merge [branch]

- Merge two branches.
- Example: `git merge new-feature`

## 4. git checkout [branch]

- Switch branches or restore working tree files.
- Example: `git checkout main`

# Advanced Commands

## 1. git stash

- Temporarily store changes.
- Example: `git stash`

## 2. git rebase [branch]

- Reapply commits on top of another base tip.
- Example: `git rebase main`

## 3. git log

- Show commit logs.
- Example: `git log`

## 4. git diff

- Show changes between commits, commit and working tree, etc.
- Example: `git diff`

# Handling Remotes

## 1. git remote add [name] [url]

- Add a remote repository.
- Example: `git remote add origin https://github.com/user/repo.git`

## 2. git fetch [remote]

- Download objects and refs from another repository.
- Example: `git fetch origin`

# Repository Management

## 1. Creating a Repository

- Done through the GitHub interface: 'New repository' button.

## 2. Forking a Repository

- Click the 'Fork' button on the repository's page.

## 3. Managing Branches

- Use `git branch`, `git checkout`, and GitHub's interface.

# Collaborating

## 1. Pull Requests

- Create: 'New pull request' on GitHub.
- Review: Comment and approve on the PR page.

## 2. Issues

- Create an issue using the 'New issue' button.

## 3. Project Boards

- Manage through the 'Projects' tab in a repository.

# Best Practices

## 1. Regular Commits

- Commit often with `git commit -m "message"`.

## 2. Clear Commit Messages

- Example: `git commit -m "Add user login functionality"`.

## 3. Code Reviews

- Review code in pull requests before merging.

## 4. Branching Strategies

- Use feature branches, merge into 'develop' or 'main'.

# Markdown

## 1. README.md

- Basic syntax: ``Title``, ``Subtitle``, ``List item``

# GitHub Workflow: Step-by-Step Guide

## 1. Create a New Repository

- On GitHub: Click the '+' icon on the top right and select 'New repository'.
- Name Your Repository: Choose a meaningful name and provide a brief description.
- Initialize with README: Optionally, initialize the repository with a README file.

## 2. Clone the Repository

- Clone to Your Machine: Use the `git clone [url]` command.
- Example: `git clone https://github.com/yourusername/your-repository.git`

## 3. Create a New Branch

- Create Branch Locally: Use `git checkout -b [branch-name]`.
- Example: `git checkout -b feature-branch`
- Push the Branch to GitHub: Use `git push -u origin [branch-name]`.

## 4. Make Changes Locally

- Edit Files: Make changes or additions to your project files.
- Add Changes: Use `git add .` to add all changed files to the staging area.
- Commit Changes: Use `git commit -m "Commit message"` to commit your changes.

## 5. Push Changes to GitHub

- Push Your Branch: Use `git push origin [branch-name]`.

- Example: `git push origin feature-branch`

## 6. Create a Pull Request (PR)

- On GitHub: Navigate to your repository and click 'Compare & pull request' next to your branch.
- Describe Your Changes: Add a meaningful title and description to your PR.
- Create the PR: Click 'Create pull request'.

## 7. Review and Merge the Pull Request

- Code Review: Team members review the changes, discuss, and make suggestions.
- Approve the PR: Once reviewed and approved, the PR can be merged.
- Merge the PR: Click 'Merge pull request' and confirm.

## 8. Pull Merged Changes

- Update Local Repository: Switch to your main branch (`git checkout main`) and use `git pull origin main` to get the latest changes.

## 9. Delete the Feature Branch (Optional)

- Delete Remote Branch: On GitHub, you can delete the branch post-merge.
- Delete Local Branch: Use `git branch -d [branch-name]`.

## 10. Repeat the Process

- For new features or changes, start again from step 3.