

Git Cheatsheet

Basic Commands

Configuration

...

Set user name and email

```
git config --global user.name "Your Name"
```

```
git config --global user.email "your.email@example.com"
```

Check configuration

```
git config --list
```

...

Repository

...

Initialize a new repository

```
git init
```

Clone an existing repository

```
git clone <repository_url>
```

...

Staging and Committing

...

Check status of the repository

git status

Add files to the staging area

git add <file>

git add . # Add all files

Commit changes

git commit -m "Commit message"

Add and commit in one step

git commit -am "Commit message"

...

Branching

...

List all branches

git branch

Create a new branch

git branch <branch_name>

Switch to a branch

git checkout <branch_name>

Create and switch to a branch

git checkout -b <branch_name>

Delete a branch

```
git branch -d <branch_name>
```

...

Merging

...

Merge a branch into the current branch

```
git merge <branch_name>
```

...

Remote Repositories

...

Add a remote repository

```
git remote add origin <repository_url>
```

List remote repositories

```
git remote -v
```

Push changes to remote repository

```
git push origin <branch_name>
```

Pull changes from remote repository

```
git pull origin <branch_name>
```

...

Logs and History

...

View commit history

git log

git log --oneline # Compact view

Show changes in a specific commit

git show <commit_hash>

...

Undoing Changes

...

Unstage a file

git reset <file>

Undo the last commit (keep changes)

git reset --soft HEAD~1

Undo the last commit (discard changes)

git reset --hard HEAD~1

...

Tags

...

List all tags

git tag

Create a new tag

```
git tag <tag_name>
```

```
# Push tags to remote
```

```
git push origin <tag_name>
```

```
...
```

```
### Stash
```

```
...
```

```
# Save changes to stash
```

```
git stash
```

```
# List stashed changes
```

```
git stash list
```

```
# Apply stashed changes
```

```
git stash apply
```

```
# Drop stash
```

```
git stash drop
```

```
...
```

```
### Other Commands
```

```
...
```

```
# Compare changes
```

```
git diff
```

```
git diff <branch1> <branch2>
```

Check current branch

git branch --show-current

Rename default branch

git branch -m <new_name>

...

Helpful Tips

- Use `git status` frequently to check the state of your repository.
- Commit often with meaningful messages.
- Always pull changes from remote before pushing.
- Create branches for new features or experiments.
