

# Detailed Git Cheat Sheet with Examples

## 1. Git Configuration

### Set username:

```
git config --global user.name "John Doe"
```

Sets your name for all repositories on your system.

### Set email:

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

Sets your email for all repositories. Important for associating commits.

### View config:

```
git config --list
```

Lists all the Git configurations set in your environment.

## 2. Starting a Repository

### Initialize repo:

```
git init
```

Creates a new Git repository in the current directory.

### Clone repo:

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

Clones an existing repository from GitHub to your local machine.

## 3. Staging and Committing

### Add file:

```
git add index.html
```

Stages a file (index.html) for the next commit.

### Add all:

```
git add .
```

Stages all modified and new files.

### Commit changes:

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

Commits staged changes with a message describing the changes.

## 4. Branching

### New branch:

```
git branch feature-login
```

Creates a new branch named 'feature-login'.

### Switch branch:

```
git checkout feature-login
```

Switches to the branch 'feature-login'.

### Create & switch:

```
git checkout -b feature-login
```

Creates and switches to 'feature-login' in one step.

## 5. Merging and Rebasing

### Merge branch:

```
git merge feature-login
```

Merges changes from 'feature-login' into the current branch.

### Abort merge:

```
git merge --abort
```

Stops the merge process and reverts to the state before merging.

### Rebase:

```
git rebase main
```

Moves the entire branch to begin on the tip of 'main' branch.

## 6. Remote Repositories

### Push branch:

```
git push origin main
```

Pushes local 'main' branch to remote 'origin'.

### Set upstream:

```
git push -u origin main
```

Pushes and sets the default upstream tracking for future pushes.

## 7. Undo Commands

### Undo add:

```
git reset HEAD <file>
```

Unstages the file but keeps your changes in the working directory.

### **Undo commit (keep changes):**

```
git reset --soft HEAD~1
```

Removes the last commit but retains your changes staged.

### **Undo commit (unstage):**

```
git reset --mixed HEAD~1
```

Removes the last commit and unstages the changes.

### **Undo commit (discard all):**

```
git reset --hard HEAD~1
```

Removes the last commit and all related changes permanently.

### **Revert commit:**

```
git revert <commit-hash>
```

Creates a new commit that reverses the changes made by the commit.

## **8. Logs and History**

### **View log:**

```
git log
```

Shows the commit history with details.

### **Online log:**

```
git log --oneline
```

Displays commits in a compact form.

### **Diff changes:**

```
git diff
```

Shows what has changed but not yet staged.