

Git Advanced Hindi-Friendly Cheat Sheet

Introduction

Yeh Git Cheat Sheet un developers ke liye hai jo roz Git ka use karte hain.

Isme basic se lekar advanced level tak ke workflows (main/dev/feature/release) aur commands include kiye gaye hain.

1. Clone and Setup

```
$ git clone <repo-url>
```

-> Repository ko local par clone karne ke liye

```
$ cd project-folder
```

-> Project folder ke andar jaane ke liye

2. Branching

```
$ git checkout -b dev
```

-> Nayi dev branch create karke switch hona

```
$ git push origin dev
```

-> Dev branch ko GitHub par bhejna

```
$ git checkout main
```

-> Wapas main branch par switch hona

3. Feature Branch Workflow

```
$ git init
```

-> git init ek basic aur important Git command hai jo kisi bhi naye project ko Git repository banata hai.

```
$ git checkout -b feature/login
```

-> Nayi feature branch banana

```
$ git add .
```

-> Sabhi file changes ko stage karna

```
$ git commit -m "login feature added"
```

-> Changes ko commit karna

```
$ git push origin feature/login
```

-> Feature branch ko push karna

```
$ git checkout dev
```

-> Dev branch par switch hona

```
$ git merge feature/login
```

Git Advanced Hindi-Friendly Cheat Sheet

-> Feature branch ko dev me merge karna

4. Pull Latest Code

```
$ git pull origin dev
```

-> GitHub se latest dev branch ka code lana

5. Merge dev to main

```
$ git checkout main
```

-> Main branch par switch hona

```
$ git pull origin main
```

-> Main ka latest version pull karna

```
$ git merge dev
```

-> Dev branch ko main me merge karna

```
$ git push origin main
```

-> Main branch ko GitHub par push karna

6. Restore / Stash

```
$ git restore filename
```

-> Ek specific file ka change hata dena

```
$ git restore .
```

-> Saare unstaged changes ko hata dena

```
$ git stash
```

-> Changes ko temporary save karna

```
$ git stash pop
```

-> Stashed changes ko wapas lana

7. Logs and Info

```
$ git status
```

-> Status dekhna - kya kya change hua hai

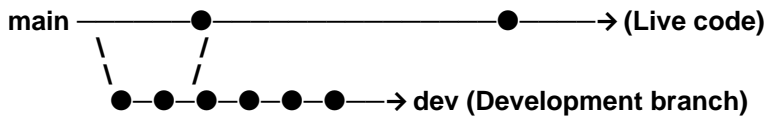
```
$ git log --oneline
```

-> Short commit history dekhna

Git Advanced Hindi-Friendly Cheat Sheet

```
$ git remote -v
```

-> Remote URLs dekhna



1. Git Repository Initialize karo (agar nahi kiya hai): `git init`

2. Remote GitHub Repository banao: `my-react-app`

3. Remote URL add karo: `git remote add origin https://github.com/your-username/my-react-app.git`
Replace your-username aur repo name se.

4. Files ko Git mein add karo:
`git add .`

5 : First Commit karo: `git commit -m "Initial commit"`

6. Main Branch par Push karo: `git branch -M main`
`git push -u origin main`

Extra Tip:

Agar aap `.gitignore` file nahi banate ho, toh `node_modules` folder bhi push ho sakta hai (jo avoid karna chahiye).

```
# .gitignore
node_modules
build
.env
```

```
git add .
git commit -m "Your message"
git push
```

Option 1: Existing Remote URL ko Replace karo (Recommended) :
`git remote set-url origin https://github.com/Kushal-Kamble/react-may`
{ Ye command purani remote URL ko nayi URL se replace kar dega. }

Check karne ke liye (optional): `git remote -v`

Ye dikha dega ki kaunsi URL currently "origin" remote ke saath linked hai.

Option 2: Purani Remote ko Hata Kar Naya Add Karna
Agar aap pura hata kar dobara add karna chahte ho:

```
git remote remove origin
git remote add origin https://github.com/Kushal-Kamble/react-may
```

Uske baad aap normal flow continue kar sakte ho: `git branch -M main`
`git push -u origin main`

git branch -m master main

git push -u origin main

▲ Pehle -m se rename hoga, phir -u se push hoke tracking set ho jayega.