

Git & GitHub Cheat Sheet in Simple Terms

Webgurukul IT Training & Placement Institute – Masterclass

1. Introduction to Git & GitHub

What is Git?

Git is a version control system that helps developers track changes in their code and collaborate with teams.

What is GitHub?

GitHub is a cloud-based platform that hosts Git repositories, allowing multiple developers to work together.

Why Use Git & GitHub?

- ✔ Track changes in code over time
- ✔ Work with teams on the same project
- ✔ Easily rollback to previous versions
- ✔ Backup code securely on GitHub

2. Installing & Configuring Git

1. Install Git

Download & install Git from: git-scm.com

2. Configure Git (First Time Setup)

```
git config --global user.name "Your Name"
git config --global user.email 'your-email@example.com'
```

3. Check Git Configuration

```
git config --list
```

3. Git Basic Commands

Initialize a New Git Repository

```
git init
```

Check the Status of the Repository

```
git status
```

Add Files to Staging Area

```
git add filename    # Add a specific file
git add .            # Add all changes
```

Commit Changes

```
git commit -m "Commit message here"
```

View Commit History

```
git log
```

4. Working with GitHub

1. Create a New Repository on GitHub

- 1 Go to GitHub > New Repository
- 2 Give it a name & click Create

2. Connect Local Git Repository to GitHub

```
git remote add origin https://github.com/your-username/repository-name.git
git branch -M main
git push -u origin main
```

3. Clone an Existing Repository

```
git clone https://github.com/your-username/repository-name.git
```

5. Git Branching

Check Current Branch

```
git branch
```

Create a New Branch

```
git branch new-feature
```

Switch to Another Branch

```
git checkout new-feature
```

OR

```
git switch new-feature
```

Create & Switch to a New Branch

```
git checkout -b new-feature
```

OR

```
git switch -c new-feature
```

Merge a Branch into Main

```
git checkout main
git merge new-feature
```

Delete a Branch

```
git branch -d new-feature
```

6. Git Pull & Push Commands

Pull Latest Changes from GitHub

```
git pull origin main
```

Push Changes to GitHub

```
git push origin main
```

Push a New Branch to GitHub

```
git push -u origin new-feature
```

7. Undoing Changes in Git

Undo Changes in Working Directory

```
git checkout -- filename
```

Undo Last Commit (Soft Reset – Keeps Changes)

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

Undo Last Commit (Hard Reset – Deletes Changes)

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

8. Git Ignore & Git Stash

Ignoring Files with .gitignore

Create a .gitignore file and add:

```
node_modules/
.env
.idea/
```

Temporarily Save Uncommitted Changes (Stash)

```
git stash
```

Apply Stashed Changes

```
git stash pop
```

9. Forking & Pull Requests (GitHub Collaboration)

1. Fork a Repository

- ✔ Go to GitHub and click Fork on a repository

2. Clone the Forked Repository

```
git clone https://github.com/your-username/forked-repo.git
```

3. Create a New Branch

```
git checkout -b feature-branch
```

4. Push Changes to Your Fork

```
git push origin feature-branch
```

5. Create a Pull Request (PR)

- ✔ Go to GitHub > Click **New Pull Request** > Select Branch & Submit

10. Resolving Merge Conflicts

Check for Merge Conflicts

```
git status
```

Resolve the Conflict Manually in the File

Example Conflict:

```
<<<<<< HEAD
This is from main branch.
=====
This is from feature branch.
>>>>>> feature-branch
```

- ✔ Go to GitHub > Click **New Pull Request** > Select Branch & Submit

```
git add filename
git commit -m "Resolved merge conflict"
```

11. GitHub Actions (Automation & CI/CD)

1. Create a GitHub Actions Workflow

- ✔ Go to GitHub > Actions > New Workflow
- ✔ Example `deploy.yml` for automatic deployment:

```
name: Deploy App

on: push

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Run Deployment Script
        run: echo "Deploying App..."
```

12. Git Best Practices

- ✔ Write meaningful commit messages
- ✔ Use branches for new features (`feature-branch`)
- ✔ Pull latest changes before pushing
- ✔ Regularly push code to GitHub for backup